Workload Scheduler
Version 8.6 (Revised June 2012)

Planning and Installation

IBM
Workload Scheduler
Version 8.6 (Revised June 2012)

Planning and Installation

IBM
Note

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

This edition applies to version 8, release 6, modification level 0 Fix pack 1 of IBM Tivoli Workload Scheduler (program number 5698-WSH) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC32-1273-10.


US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Contents

Figures .......................................................... v

Tables .............................................................. vii

About this publication ................................. ix
  What is new in this release ................................. ix
  Who should read this publication ......................... ix
  Publications .................................................... ix
  Accessibility .................................................. x
  Tivoli technical training ..................................... x
  Support information ........................................ x

Part 1. Planning .............................................. 1
  Network planning ............................................. 1
    Tivoli Workload Scheduler environment .............. 1
  Planning the environment .................................... 7
    Planning domains .......................................... 15
    Workstation classes ....................................... 21
    Time zone considerations .................................. 21

Part 2. Tivoli Workload Scheduler ....................... 23
  Preparing for installation ................................. 23
    Installation overview ..................................... 23
    Checking prerequisites (UNIX and Linux) ............... 24
    Installation considerations ............................... 24
    Installation media ......................................... 26
    Instances of Tivoli Workload Automation ............. 27
    Relational database management systems ............... 29
    Selecting your installation method ..................... 30
    Installation log files ..................................... 32
    Tivoli Workload Scheduler user ......................... 34
      Windows services ......................................... 36
  Creating or upgrading the Tivoli Workload Scheduler database tables before installing or upgrading .................................................. 36
    Creating or upgrading the database tables if you are using DB2 ........................................ 37
    Creating or upgrading the database tables if you are using Oracle .................................... 47
  Installing ..................................................... 59
    User authorization requirements ...................... 59
    Installing DB2 ............................................ 59
    Using the installation wizard ......................... 60
    Performing a silent installation .................... 84
    Installing agents using twsinstall .................. 88
    Installing agents using Software Distribution ..... 96
    Installing the Job Brokering Definition Console .... 105
    Installing the additional plug-ins by using the Tivoli Workload Scheduler for Additional Plug-ins ............................................. 106
  Upgrading ..................................................... 112
    Engine coexistence and upgrade notes ................ 112
    User authorization requirements ...................... 113
    Checking prerequisites (UNIX and Linux) ............ 114
    Upgrading a master domain manager or backup master domain manager instance ............... 114
    Upgrading agents and domain managers ................ 141
    Upgrading a command line client ..................... 155
    Upgrading when there are corrupt registry files .... 156
    Adding a feature .......................................... 158
  Configuring .................................................... 159
    Setting the environment variables .................... 159
    Configuring a master domain manager ................. 160
    Configuring a backup master domain manager ...... 161
    Configuring a domain manager ........................... 161
    Configuring a backup domain manager ............... 162
    Configuring a dynamic domain manager ............. 163
    Configuring a backup dynamic domain manager ....... 163
    Configuring a fault-tolerant agent .................... 164
    Configuring a dynamic agent ............................. 165
    Configuring a command line client ................. 165
    Configuring WebSphere Application Server ............ 165
    Adding a feature .......................................... 165
    Enabling dynamic scheduling after installation .... 168

Part 3. Dynamic Workload Console ............... 245
  Preparing ..................................................... 245

© Copyright IBM Corp. 1999, 2012
## Figures

1. Graphical overview of Tivoli Workload Scheduler environment to run static workload | 2
2. Graphical overview of Tivoli Workload Scheduler dynamic environment | 4
3. Distributed workload environment with static scheduling capabilities | 8
4. Distributed workload environment with dynamic scheduling capabilities | 10
5. Distributed workload environment with static and dynamic scheduling capabilities | 12
6. Workload environment integrated with external systems | 14
7. Distributed-driven workload environment for z/OS | 15
8. Single domain topology | 18
9. Single domain topology on multiple sites | 19
10. Multiple domain topology | 20
11. Upgrade procedure flowchart | 116
12. Wizard panel after an installation failure | 180
13. Step List window showing a failed step | 181
14. Step status tab | 183
15. Step properties tab | 183
16. Step output tab | 185
### Tables

1. Features partially or not supported for dynamic scheduling .......................... 11
2. Symbolic link options .............................................................................. 26
3. Installing into an existing instance of Tivoli Workload Automation ................. 28
4. Installation log files ................................................................................. 32
5. DB2 Setup files ....................................................................................... 60
6. Response files .......................................................................................... 84
7. List of parameters to install language packs .............................................. 104
8. Options to perform a silent installation .................................................... 109
9. Installation log files .................................................................................. 111
10. Upgrade availability for Tivoli Workload Scheduler components ............... 112
11. Direct upgrade scenario .......................................................................... 117
12. Parallel upgrade scenario ....................................................................... 118
13. Uninstallation log files ............................................................................ 176
14. Options to perform a silent uninstallation ................................................ 177
15. Default InstallAnywhere error messages .................................................. 192
16. InstallAnywhere error messages for additional plug-ins ............................ 193
17. Dynamic Workload Console response files .............................................. 223
18. Installing into an existing instance of Tivoli Workload Automation ............. 247
19. Installation log files ................................................................................. 249
20. Dynamic Workload Console response files ............................................. 254
21. Objects downloaded by the tutorial utility ............................................ 284
22. List of scheduling scenarios .................................................................. 285
23. Registry file attributes .......................................................................... 291
24. Tivoli Workload Scheduler response file properties .................................. 292
25. Dynamic Workload Console response file properties ............................... 303
26. Job Brokering Definition Console response file properties ........................ 308
About this publication

This IBM® Tivoli® Workload Scheduler Planning and Installation provides information for planning, installing, migrating, and configuring an IBM Tivoli Workload Scheduler network.

What is new in this release

For information about the new or changed functions in this release, see Tivoli Workload Automation: Overview.

For information about the APARs that this release addresses, see the Tivoli Workload Scheduler Download Documents at http://www.ibm.com/support/docview.wss?rs=672&uid=swg24027501 and Dynamic Workload Console Download Documents at http://www.ibm.com/support/docview.wss?rs=672&uid=swg24029125.

Who should read this publication

This guide is intended for the following audience:

- Tivoli Workload Scheduler IT administrators who plan for and install the network
- Dynamic Workload Console IT administrators who plan for and install the product
- Specialists who plan the network topology
- IT administrators who install the network
- System architects

Publications

Full details of Tivoli Workload Automation publications can be found in Tivoli Workload Automation: Publications. This document also contains information on the conventions used in the publications.

A glossary of terms used in the product can be found in Tivoli Workload Automation: Glossary.

Both of these are in the Information Center as separate publications.

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.

Tivoli technical training

For Tivoli technical training information, refer to the following IBM Tivoli Education website:

http://www.ibm.com/software/tivoli/education

Support information

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

Online

Go to the IBM Software Support site at http://www.ibm.com/software/support/probsub.html and follow the instructions.

IBM Support Assistant

The IBM Support Assistant (ISA) is a free local software serviceability workbench that helps you resolve questions and problems with IBM software products. The ISA provides quick access to support-related information and serviceability tools for problem determination. To install the ISA software, go to http://www.ibm.com/software/support/isa.

Troubleshooting Guide

For more information about resolving problems, see the problem determination information for this product.

For more information about these three ways of resolving problems, see “Support information.”

For more information about these three ways of resolving problems, see the appendix on support information in Tivoli Workload Scheduler: Troubleshooting Guide, SC32-1275.
Part 1. Planning

Network planning

This section provides information to help you plan your Tivoli Workload Scheduler network.

Tivoli Workload Scheduler environment

A Tivoli Workload Scheduler network consists of a set of linked workstations on which you perform job processing. A network is composed of one or more domains, each having a domain manager workstation acting as a management hub, and one or more agent workstations.

Using Tivoli Workload Scheduler you can run your workload in one of the following ways:

Statically
To run existing job types, for example docommand and scripts on specific workstations of fault-tolerant agents or standard agent type.

Dynamically
To run existing job types and job types with advanced options, allowing the product to assign it to the workstation that best meets both the hardware and software requirements needed to run it.

Job types with advanced options are both those supplied with the product and the additional types implemented through the custom plug-ins. For example, those supplied with the product are DB2®, file transfer, and web services. Those implemented through the custom plug-ins are the ones you developed using the Integration Workbench of the Software Development Kit (SDK).

Depending on how you want to run your workload you have to install and configure different components in your network.

Figure 1 on page 2 gives a graphical overview of a typical Tivoli Workload Scheduler environment to run static workload:
In Figure 1 the master domain is shown with the principle components to run workload statically, 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:

To run your workload statically install the following components:

**Master domain manager**

The master domain manager is the highest level workstation of a Tivoli Workload Scheduler network. It contains or connects to the relational database that stores scheduling object definitions. It creates or updates a production file when the plan is created or extended and then distributes the file 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.

**Domain manager**
Install this component if you need a multi-domain network and you want to manage workload by assigning it to a predefined workstation that is to run your workload statically. In a multi-domain network all domains below the master domain have fault-tolerant agents configured to be a domain manager to manage the workstations in its domain. A domain manager can manage fault-tolerant, standard, and extended agents. Each domain manager is a fault-tolerant agent in the domain of the next higher level. To define a domain manager, install a fault-tolerant agent on your workstation and then define it as **manager** in the workstation definition.

**Backup domain manager**
Install this component if you want a backup to your domain manager. If your domain manager experiences problems, you can configure any fault-tolerant agent as the domain manager and 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 Tivoli Workload Scheduler master domain manager. Install agents by choosing the agent installation from the DVD or by downloading the elmage using the Passport Advantage® Online website. After installing the agent, you define its type by using the workstation definition.

**Fault-tolerant agent**
An 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.

**Extended agent**
Extended agents are logical definitions (hosted by a physical workstation) used to extend job processing to selected applications (SAP R/3, Oracle E-Business Suite, PeopleSoft, and z/OS®). For information about installing an extended agent, see **Tivoli Workload Scheduler for Applications: User’s Guide**.

**Note:** All agents with special roles (master domain manager, backup master domain manager, domain manager, backup domain manager) can also work as fault-tolerant agents with jobs scheduled on them.

Figure 2 on page 4 gives a graphical overview of a typical Tivoli Workload Scheduler environment to run dynamic workload:
In Figure 2, the master domain is shown with the principle components to run workload dynamically, and two levels of dynamic 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.

If you want to run your workload dynamically install the following components:

**Master domain manager**

The master domain manager is the highest level workstation of a Tivoli Workload Scheduler network. It contains or connects to the relational database that stores scheduling object definitions. It creates or updates a production file when the plan is created or extended and then distributes the file 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.

Dynamic Domain manager
Install this component if you need a multi-domain network and you want to manage your workload both statically that dynamically. All domains below the master domain have dynamic domain managers to manage the workstations in its domain. Each dynamic domain manager is an agent in the domain of the next higher level. To define a dynamic domain manager, install a dynamic domain manager and then perform the "Configuring a dynamic domain manager" on page 163 procedure.

Backup dynamic domain manager
Install this component if you want a backup to your dynamic domain manager. 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 Tivoli Workload Scheduler master domain manager. Install agents by choosing the agent installation from the DVD or by downloading the eImage using the Passport Advantage Online website.

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 provides high availability and load balancing potentialities to your environment and well suite virtualized environments.

When a job is submitted, either as part of a job stream in the plan or through ad hoc submission, Tivoli Workload Scheduler 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.

Run both existing job types and job types with advanced options
It can run:
• Existing job types. For example docommand and scripts.
• Job types with advanced options, both those supplied with the product and the additional types implemented through the custom plug-ins. For example, those supplied with the product are DB2, file transfer, and web services. Those implemented through the custom plug-ins are the ones you developed using the
Integration Workbench of the Software Development Kit (SDK). To run these job types you must also install the Java™ runtime.

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 runtime.

After installing the agent, you define its type by using “Configuring a dynamic agent” on page 165.

Note: Dynamic agents must be directly connected to the master domain manager or to the dynamic domain manager.

Extended agent
Extended agents are logical definitions (hosted by a physical workstation) used to extend job processing to selected applications (SAP R/3, Oracle E-Business Suite, PeopleSoft, and z/OS). For information about installing an extended agent, see Tivoli Workload Scheduler for Applications Tivoli Workload Scheduler for Applications: User’s Guide.

Tivoli Workload Scheduler interfaces
Tivoli Workload Scheduler includes the following user interfaces from which you manage your production environment:

Master domain manager command line
The master domain manager command line is installed automatically when you install the master domain manager. This command line interface is run only from the workstation serving as the master domain manager. From the command line, you can administer the master specific binaries and options. A backup master domain manager command line also exists on the backup master domain manager.

Dynamic Workload Console
The web-based interface for creating, modifying, monitoring, controlling, and deleting Tivoli Workload Scheduler objects. You can interface with the console from any system in the network where a supported web browser is installed.

Command line client
A component of Tivoli Workload Scheduler 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)

Tivoli dynamic workload broker command line
Installed and configured automatically when you select to enable the dynamic scheduling capability at master installation time. It includes commands to directly submit and manage jobs for dynamic scheduling, manage job JSDL definitions and resources, and more. See Tivoli Workload Scheduler: Scheduling Workload Dynamically for reference.

Job Brokering Definition Console
A structured editing tool that you use to create and modify Job Submission
Description Language (JSDL) files. These files are saved in the Job Repository as job definitions and become available for submission. The JSDL files adhere to the XML syntax and semantics as defined in the JSDL schema. For more information, see the Tivoli Workload Scheduler: User’s Guide and Reference, SC32-1274.

Additionally, Tivoli Workload Automation contains the z/OS Connector, which is a component that connects IBM Tivoli Workload Scheduler for z/OS and the Dynamic Workload Console. For more information, see Tivoli Workload Scheduler for z/OS: Planning and Installation Guide.

For a more detailed description of the Tivoli Workload Scheduler components, see Tivoli Workload Automation: Overview.

Planning the environment
This section describes some of the typical installation scenarios for Tivoli Workload Automation products and components. These typical scenarios show how to deploy specific solutions on the minimum possible system resources.

Distributed workload environment with static scheduling capabilities
Use this configuration to run workload statically across your distributed network. Figure 3 on page 8 shows the system resources needed to install a fully-working Tivoli Workload Scheduler environment for managing your distributed workload.
Figure 3. 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. In this configuration, you can choose whether or not to add the runtime environment for Java jobs to the agent. The runtime 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 Tivoli Workload Scheduler: Scheduling Workload Dynamically.

Figure 4 on page 10 shows the system resources required to install a fully working Tivoli Workload Scheduler environment for running your distributed workload dynamically.

Note: A dynamic agent can be directly connected to its master domain manager, as shown in Figure 4 on page 10 or through a dynamic domain manager as shown in Figure 5 on page 12.
Dynamic scheduling supports most of the Tivoli Workload Scheduler features for static scheduling. The Table 1 on page 11 lists some features or properties that are partially or not supported.

Figure 4. Distributed workload environment with dynamic scheduling capabilities
Table 1. Features partially or not supported for dynamic scheduling

<table>
<thead>
<tr>
<th>Feature</th>
<th>dynamic agent and Tivoli Workload Scheduler for z/OS agent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event-driven workload automation.</td>
<td>TWSObjectMonitor events supported</td>
</tr>
<tr>
<td><strong>Note:</strong> For more details about the events type, see Tivoli Workload Scheduler User’s Guide and Reference: Appendixes - Event-driven workload automation event and action definitions</td>
<td>TFileMonitor events not supported</td>
</tr>
<tr>
<td></td>
<td>TWSApplicationMonitor events not supported</td>
</tr>
<tr>
<td>File dependency</td>
<td>Not supported</td>
</tr>
<tr>
<td>Utility commands (datecalc, jobinfo etc.)</td>
<td>Not supported</td>
</tr>
</tbody>
</table>

**Distributed workload environment with static and dynamic scheduling capabilities**

Use this configuration to run workload both statically and dynamically across your distributed network. In this configuration, you can choose whether or not to add the runtime environment for Java jobs to the agent. The runtime 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 *Tivoli Workload Scheduler: Scheduling Workload Dynamically*.

Figure 5 on page 12 shows the system resources required to install a fully working Tivoli Workload Scheduler environment for running your distributed workload both statically and dynamically. Tivoli Workload Scheduler 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 5 on page 12.
For a list of features partially or not supported in a mixed environment, see Table 1 on page 11.

End-to-end workload environment
In an end-to-end environment (agent connected to the z/OS system), you can define the following types of configurations:

To run your workload statically:
Using fault-tolerant agents
Use the fault-tolerant end-to-end scheduling environment to schedule and control static workload from the mainframe to distributed systems. On the distributed system, you install fault-tolerant agents and connect them to the z/OS server. See Tivoli Workload Scheduler for z/OS: Scheduling End-to-end with Fault Tolerance Capabilities for more details.

Using Tivoli Workload Scheduler for z/OS 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 Tivoli Workload Scheduler for z/OS agents and connect them to the z/OS controller. For information about how to install it see Tivoli Workload Scheduler for z/OS: Planning and Installation Guide for information about how to use it see Tivoli Workload Scheduler for z/OS: Scheduling End-to-end with z-centric Capabilities for more details.

To run your workload dynamically:

Using Tivoli Workload Scheduler for z/OS 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 Tivoli Workload Scheduler for z/OS 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 a dynamic domain manager or backup dynamic domain manager” on page 72
- Install Tivoli Workload Scheduler for z/OS agents see Tivoli Workload Scheduler for z/OS: Planning and Installation Guide
- Use Tivoli Workload Scheduler for z/OS agents see Tivoli Workload Scheduler for z/OS: Scheduling End-to-end with z-centric Capabilities for more details.

Workload environment integrated with external systems
Use this configuration to extend Tivoli Workload Scheduler capabilities for scheduling on external applications, such as SAP R/3 and PeopleSoft using Tivoli Workload Scheduler.

Figure 6 on page 14 shows a sample environment including the agents needed to extend Tivoli Workload Scheduler scheduling capabilities on one or more external applications using Tivoli Workload Scheduler for Applications. You can install Tivoli Workload Scheduler for Applications on the master domain manager, on a fault-tolerant agents, on dynamic agents, and on Tivoli Workload Scheduler for z/OS agents.

For information about Tivoli Workload Scheduler for Applications, see the Tivoli Workload Scheduler for Applications: User’s Guide documentation.
Note: Installing Tivoli Workload Scheduler for Applications on an agent (master domain manager, domain manager, fault-tolerant agent, standard agent, dynamic agent, Tivoli Workload Scheduler for z/OS agent) is the correct deployment scenario in an end-to-end environment.

Distributed-driven workload environment for z/OS
Use this configuration to submit from the Tivoli Workload Scheduler (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 6 shows the minimum system resources needed to install a distributed-driven environment, where the Tivoli Workload Scheduler 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 Tivoli Workload Scheduler distributed-Agent for z/OS, see the Tivoli Workload Scheduler: Scheduling with the Agent for z/OS documentation.

Planning domains

A Tivoli Workload Scheduler 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**
In planning your Tivoli Workload Scheduler 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 21.
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 Tivoli Workload Scheduler network at each location. Some decentralized management is possible in a stand-alone Tivoli Workload Scheduler network by mounting or sharing file systems.

Types of applications
Consider the types of applications that are run by Tivoli Workload Scheduler. 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 Tivoli Workload Scheduler 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, thereby decreasing network traffic and improving use of the domain architecture. See the Tivoli Workload Scheduler: User's Guide and Reference, SC32-1274.

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 the Tivoli Workload Scheduler: Administration Guide.

Secure Sockets Layer (SSL) or IBM Global Security Kit (GSKit) encryption
If you want to use SSL or GSKit encryption in your network, plan your domains in accordance with the protocol.

Note: If you want to be compliant with Federal Information Processing Standards (FIPS), you must use GSKit. See the Tivoli Workload Scheduler: Administration Guide.

Single domain network
A single domain network consists of a master domain manager and any number of agents. Figure 8 on page 18 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.
Single domain networks can be combined with other networks, single or multiple domain, to meet multiple site requirements. Tivoli Workload Scheduler supports internetwork dependencies between jobs running on different networks.
Example 1

Example 2

Figure 9. Single domain topology on multiple sites

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.

As Figure 10 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 Boulder domain manager is routed through its parent domain manager in Denver. If there are schedules or jobs in the Boulder
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.

**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. For information about how to set the time zone, see *Tivoli Workload Scheduler: Administration Guide*.

Time zone implementation also enables easy scheduling across multiple time zones. For a description of how the time zone works, see the *Tivoli Workload Scheduler: User’s Guide and Reference*.
Part 2. Tivoli Workload Scheduler

Preparing for installation

This chapter provides a brief overview of an installation and some specific environment considerations. It contains the following sections:

- “Installation overview”
- “Checking prerequisites (UNIX and Linux)” on page 24
- “Installation considerations” on page 24
- “Installation media” on page 26
- “Instances of Tivoli Workload Automation” on page 27
- “Relational database management systems” on page 29
- “Selecting your installation method” on page 30
- “Installation log files” on page 32
- “Tivoli Workload Scheduler user” on page 34
- “Windows services” on page 36

Installation overview

Perform the following steps to prepare to install and configure Tivoli Workload Scheduler:

1. Confirm the configuration of your network to determine what type of workstation to install. See “Network planning” on page 1.
2. Check the installation prerequisites at http://www.ibm.com/support/docview.wss?rs=672&uid=swg27019747 to verify that your system is compliant.
3. Decide if you want to use a DB2 database, or an Oracle database.
4. Run the procedure described in “Creating or upgrading the Tivoli Workload Scheduler database tables before installing or upgrading” on page 36 only when the database administrator manages all the confidential information related to the database, such as the database administrator user ID and password and the IT administrator who installs the product does not know them.
5. Decide if you want to install into an existing instance of Tivoli Workload Automation or are installing a new instance of Tivoli Workload Automation.
6. Collect the information necessary to fill in the required fields during the installation.
7. Optional, create or upgrade the Tivoli Workload Scheduler database tables before installing or upgrading. The database administrator runs this procedure only if the IT administrator who installs the product does not know all the confidential information related to the database. If the IT administrator can provide the database administrator user ID and password during the installation, the database administrator does not need to run these procedures because the installation automatically creates and upgrades the database tables.
8. Install Tivoli Workload Scheduler following the instructions provided in “Installing” on page 59.
9. Perform any configuration required for the workstation type you installed. See “Configuring” on page 159.
Checking prerequisites (UNIX and Linux)

If you are preparing to install on UNIX and Linux operating systems, Tivoli Workload Scheduler automatically runs a prerequisite check on your system. Having an environment that meets the Tivoli Workload Scheduler system requirements ensures that an installation succeeds without any delays or complications.

**Note:** The prerequisite check is not available if you are using the Software Distribution installation method and is only available on UNIX and Linux operating systems.

The prerequisite check verifies that:
- The operating system is supported for the product.
- The necessary engine software patch levels are installed.
- There is enough permanent and temporary disk space.
- There is enough memory and virtual memory swap space.
- The necessary kernel parameters are correctly configured.

**Note:** The prerequisite check only verifies that the environment meets the requirements of the Tivoli Workload Scheduler. It does not check the installation requirements for other components, such as DB2.

If any of these checks fails Tivoli Workload Scheduler displays a notification of the requirement that was not met, and you can pause the installation to resolve the problem. Without this check, the software might install but then fail to work.

If you are installing Tivoli Workload Scheduler using the installation wizard, silent installation, or twsinst script, you can control whether the prerequisite check stops the installation only when it encounters blocking errors or when it encounters any type of error or warning. If you want the installation to stop for any error or warning, perform the following when you launch the installation:

**Installation wizard**

Enter `SETUP.bin -W checkPrerequisites.stopOnCheckPrereq=true`.

**Silent installation**

Specify the parameter `-W checkPrerequisites.stopOnCheckPrereq=true` in the response files you use for silent installations.

**twsinst**

Specify the parameter `-stoponcheckprereq`.

Details of the prerequisite check results are available in the installation log (see "Installation log files" on page 248). In the installation log you find details of the blocking and non-blocking errors, as well as any warnings.

For a detailed list of supported operating systems and product prerequisites, see [http://www.ibm.com/support/docview.wss?rs=672&uid=swg27019747](http://www.ibm.com/support/docview.wss?rs=672&uid=swg27019747)

## Installation considerations

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. This is required for IBM WebSphere Application Server authentication.
• If <TWS_user> is a domain user, the user performing the installation must be a domain administrator.

Installing on UNIX and Linux

If you are installing on HP-UX operation systems, check that the MAXDSIZ parameter is set to a minimum of 128 MB.

If you get an error message indicating insufficient space for the installation wizard temporary data in the default /tmp directory, you can launch the installation wizard with the -is flag and set an alternative temporary directory. For example:

SETUP.sh [-is:tempdir <temporary_directory>]

For additional information about disk and space requirements for the installation, see "Checking prerequisites (UNIX and Linux)" on page 24.

Installing with DB2 or installing several computers from a mounted shared directory

The installation DVDs include two types of installation scripts:
• <operating_system>/SETUP.bin
• SETUP.sh

SETUP.sh makes a local copy of the installation media in /tmp/_twscd. If you use this method, ensure that there is adequate space in /tmp. For more information, see the Tivoli Workload Scheduler System Requirements Document at [http://www.ibm.com/support/docview.wss?rs=672&uid=swg27019747].

Choosing language settings and national characters

If you want to use characters of a specific language locale, the language you choose for the installation wizard must match the language locale settings of the workstation on which you are installing. You cannot use national characters in the installation path of a master domain manager or backup master domain manager. Additionally, you cannot add a distributed connector to an agent that has national characters in its installation path.

Installation errors

If the installation ends with errors, do not use the Close icon on the top right to exit the session because this prevents the creation of installation summary log file. Complete the installation even if it contains errors by clicking Next until you reach the last panel and then Finish.

Performing silent installations

When you install the latest version of Tivoli Workload Scheduler, you can create a response file based on the parameters of the initial installation. You can then use this customized response file to run silent installations using the same parameters. Before running the initial installation, you might want to consider this feature. For more information, see "Performing a silent installation" on page 84.

Mapped drives

When you copy the image of a specific operating system onto the
workstation for installation using the wizard, you must copy the complete contents of the DVD to the drive from where you run your installation. When the drive is a UNC mapped drive, the remote path must be mapped to a drive on the installation workstation. For a complete list of the supported operating systems and their prerequisites, see the Tivoli Workload Scheduler System Requirements Document [http://www.ibm.com/support/docview.wss?rs=672&uid=swg27019747].

Remote installation
You cannot install Tivoli Workload Scheduler on a Windows workstation from a remote Samba-mounted file system.

Installing for end-to-end scheduling
If you are installing Tivoli Workload Scheduler 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 Tivoli Workload Scheduler Scheduling End-to-end.

Create symbolic links
UNIX and Linux. The installation wizard installs all executable files in its own .bin directory. Before running any Tivoli Workload Scheduler 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 shows the binary paths and the symbolic links.

<table>
<thead>
<tr>
<th>TWS binary path</th>
<th>Symbolic link</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;TWS_home&gt;/bin/at</td>
<td>/usr/bin/mat</td>
</tr>
<tr>
<td>&lt;TWS_home&gt;/bin/batch</td>
<td>/usr/bin/mbatch</td>
</tr>
<tr>
<td>&lt;TWS_home&gt;/bin/datecalc</td>
<td>/usr/bin/datecalc</td>
</tr>
<tr>
<td>&lt;TWS_home&gt;/bin/jobstdl</td>
<td>/usr/bin/jobstdl</td>
</tr>
<tr>
<td>&lt;TWS_home&gt;/bin/maestro</td>
<td>/usr/bin/maestro</td>
</tr>
<tr>
<td>&lt;TWS_home&gt;/bin/mdemon</td>
<td>/usr/bin/mdemon</td>
</tr>
<tr>
<td>&lt;TWS_home&gt;/bin/morestdl</td>
<td>/usr/bin/morestdl</td>
</tr>
<tr>
<td>&lt;TWS_home&gt;/bin/muser</td>
<td>/usr/bin/muser</td>
</tr>
<tr>
<td>&lt;TWS_home&gt;/bin/parms</td>
<td>/usr/bin/parms</td>
</tr>
</tbody>
</table>

Installation media
Depending on the operating system, the installation DVD contains some or all the following directories:

TWS Contains the files necessary to install Tivoli Workload Scheduler
TDWC Contains the files necessary to install the Dynamic Workload Console
DB2 Contains the files necessary to install DB2
DB2_activation Contains the files necessary for DB2 activation
Integrations
Contains the files necessary to integrate Tivoli Workload Scheduler with other Tivoli products

Launchpad
Contains the launch pad code

Integration Workbench
Contains the files necessary to install Tivoli Workload Scheduler Integration Workbench

JBDC
Contains the files necessary on a Windows or Linux system for the Job Brokering Definition Console

For a complete list of the supported operating systems, see the Tivoli Workload Scheduler downloadable documentation at [http://www.ibm.com/support/docview.wss?rs=672&uid=swg24027501](http://www.ibm.com/support/docview.wss?rs=672&uid=swg24027501).

**Instances of Tivoli Workload Automation**

Tivoli Workload Scheduler installs files for the TWS_user in the directory selected path\TWS\ and selected path\eWAS\, where selected path is the installation location.

On Windows operating systems, the default installation location for a new installation is c:\Program Files\IBM\TWA\. On UNIX operating systems, the default installation location is /opt/IBM/TWA/.

On Linux operating systems, the product is installed into the directory chosen during installation. The default installation location is /opt/ibm/TWA/.

Each instance of a Tivoli Workload Scheduler component can exist only once in a Tivoli Workload Automation directory. Multiple instances of the product can be installed on a single workstation only if a unique TWS_user and installation path are used to create a separate instance.

Each instance of Tivoli Workload Automation can contain the following:

• One instance of the embedded IBM WebSphere Application Server on which can run:
  – One instance of a master domain manager, backup master domain manager, dynamic domain manager, backup dynamic domain manager, domain manager with distributed connector, or fault-tolerant agent with distributed connector
  – One instance of the Dynamic Workload Console
  – One instance of the Tivoli Workload Scheduler for z/OS Connector

• If no other Tivoli Workload Scheduler component (master domain manager, backup master domain manager, dynamic domain manager, backup dynamic domain manager, domain manager with distributed connector, or fault-tolerant agent with distributed connector) is installed, one instance of a domain manager or fault-tolerant agent without a distributed connector.

Only one Dynamic Workload Console can be installed on a workstation and can be installed as follows:

• In an existing Tivoli Workload Automation instance
• In a new Tivoli Workload Automation instance
Outside any Tivoli Workload Automation instance, using an existing external instance of Tivoli Integrated Portal.

If you install a new Tivoli Workload Scheduler instance onto a computer that has an existing TWA directory, a new default installation directory is created as TWA1, TWA2, and so on.

Note: In this and other manuals, the Tivoli Workload Automation instance directory is referred to as TWA_home.

For example, if you have already installed Tivoli Workload Scheduler in the /opt/ibm/TWA directory, the next attempt to install Tivoli Workload Scheduler on this workstation results in an installation directory of /opt/ibm/TWA1. However, if you originally installed the Dynamic Workload Console into the /TWA/TDWC directory, you can install a new instance of Tivoli Workload Scheduler in the /opt/ibm/TWA/TWS directory. The same situation applies to Tivoli Workload Scheduler and the Dynamic Workload Console. Only one instance of each product or component can exist in any instance of a Tivoli Workload Automation directory.

Note: Instances of Tivoli Workload Scheduler are recorded only in the registry file. Former versions of Tivoli Workload Scheduler were registered both in the registry file and in the components file.

During the installation of Tivoli Workload Scheduler, decide if you want to install into an existing instance of Tivoli Workload Automation or create a new instance.

If you are installing into an existing instance of Tivoli Workload Automation, you can install certain products or components, depending on the products or components that currently exist in that instance. The advantage of installing a product or component into an existing instance of Tivoli Workload Automation is that all of the data that is required to configure the component is already present and displayed in the wizard. In some cases, data from the existing instance is reused automatically. In other cases, data is retrieved as default values that you can choose to use or edit.

Table 3 describes the actions that you can perform in each different scenario if you are installing the full set of Tivoli Workload Scheduler components.

<table>
<thead>
<tr>
<th>If the existing Tivoli Workload Automation instance contains:</th>
<th>You can perform the following:</th>
</tr>
</thead>
<tbody>
<tr>
<td>A previous version of a Tivoli Workload Scheduler component</td>
<td>Upgrade that component.</td>
</tr>
<tr>
<td>A previous version of a Tivoli Workload Scheduler agent installed using twsinst</td>
<td>Upgrade that component using twsinst.</td>
</tr>
<tr>
<td>A Tivoli Workload Scheduler version 8.6 master domain manager or backup master domain manager</td>
<td>Take no action. It is not possible to install Tivoli Workload Scheduler in this scenario.</td>
</tr>
<tr>
<td>A Tivoli Workload Scheduler version 8.6 dynamic domain manager or backup dynamic domain manager</td>
<td>Take no action. It is not possible to install Tivoli Workload Scheduler in this scenario.</td>
</tr>
<tr>
<td>A Tivoli Workload Scheduler version 8.6 fault-tolerant agent (with no connector)</td>
<td>Add the connector feature.</td>
</tr>
</tbody>
</table>
Table 3. Installing into an existing instance of Tivoli Workload Automation (continued)

<table>
<thead>
<tr>
<th>If the existing Tivoli Workload Automation instance contains:</th>
<th>You can perform the following:</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Tivoli Workload Scheduler version 8.6 fault-tolerant agent with connector</td>
<td>Take no action. It is not possible to install Tivoli Workload Scheduler in this scenario.</td>
</tr>
<tr>
<td>A Tivoli Workload Scheduler version 8.6 dynamic agent</td>
<td>Take no action. It is not possible to install Tivoli Workload Scheduler in this scenario.</td>
</tr>
<tr>
<td>A Dynamic Workload Console version 8.6</td>
<td>Install Tivoli Workload Scheduler.</td>
</tr>
<tr>
<td>A Dynamic Workload Console version 8.6 on an existing external WebSphere Application Server</td>
<td>Take no action. It is not possible to install Tivoli Workload Scheduler in this scenario.</td>
</tr>
<tr>
<td>A Tivoli Workload Scheduler for z/OS Agent version 8.6</td>
<td>Take no action. It is not possible to install another agent in this instance.</td>
</tr>
<tr>
<td>A Tivoli Workload Scheduler for z/OS connector</td>
<td>Install Tivoli Workload Scheduler.</td>
</tr>
<tr>
<td>A current version of the command line client</td>
<td>Add a language pack feature only.</td>
</tr>
</tbody>
</table>

Relational database management systems

A relational database management system (RDBMS) is a prerequisite of the master domain manager and backup master. The RDBMS can be one of the following:

DB2 Enterprise Server Edition

A version of DB2 is bundled with the installation DVD. For information about the launchpad, see “Selecting your installation method” on page 30.

You can install DB2 in the following ways:

**Server**  
Install DB2 Server and the master domain manager on the same workstation.

**Client**  
Install DB2 Server on one workstation. DB2 client and the master domain manager on a different workstation. The advantage of this configuration is that you can easily switch between your master domain manager and your backup master, if necessary.

Oracle

Install Oracle and the master domain manager on the same computer. 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 will improve rule management performance, in particular the following queries: event_rule_instance, action_run and operator_messages. For information about event-driven workload automation, see Overview.

**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.
For supported versions, see the Tivoli Workload Scheduler System Requirements Document at [http://www.ibm.com/support/docview.wss?rs=672
&uid=swg27019747](http://www.ibm.com/support/docview.wss?rs=672&uid=swg27019747).

You must install the RDBMS prior to installing Tivoli Workload Scheduler. During the installation of Tivoli Workload Scheduler, identify the instance of the RDBMS you want to use.

**Note:** If you already have an RDBMS and want to upgrade it, you must upgrade it after you upgrade Tivoli Workload Scheduler. For information about upgrading the RDBMS, see the data maintenance chapter in the *Tivoli Workload Scheduler: Administration Guide*.

### Selecting your installation method

You can install Tivoli Workload Scheduler using one of the methods described in this section.

**Launchpad**

The *launchpad* is the starting point for installing products that are part of Tivoli Workload Automation. You can also install DB2 from the launchpad. The launchpad is included in your installation media. Using the launchpad, you can:

- Install or upgrade on or all Tivoli Workload Scheduler components
- Install or upgrade the Dynamic Workload Console
- Install or upgrade Tivoli Workload Scheduler for Applications
- Install the Tivoli Workload Scheduler for z/OS Connector
- Install DB2
- Install the Job Brokering Definition Console
- Access product information
- Keep you constantly and quickly informed about product news, updates, technotes, APARs, and fixes using the "News and Updates" feature. To use this feature you must be connected to the Internet.

The launchpad automatically accesses and runs the related installation setup file in interactive mode. Note that the installation from the launchpad can be driven and simplified according to the deployment model you chose.

The launchpad requires some additional installation prerequisites. For more information, see the Tivoli Workload Scheduler System Requirements Document at [http://www.ibm.com/support/docview.wss?rs=672&uid=swg27019747](http://www.ibm.com/support/docview.wss?rs=672&uid=swg27019747).

If you have autorun enabled, the launchpad starts automatically. If you want to start the launchpad from a mounted file system, ensure that you have write permission on it before starting the launchpad.

To start the launchpad installation program, perform the following steps:

1. From the DVD that contains the component you want to install, run the launchpad as follows:

   **Windows operating systems:**
   From the root directory of the DVD, run `launchpad.exe`.

   **UNIX operating systems:**
   a. Export the browser location to the BROWSER environment variable.
   b. From the root directory of the DVD, run `launchpad.sh`.
The launchpad opens.

2. In the launchpad, click to install the configuration of Tivoli Workload Scheduler that you want. The related installation program starts. To proceed with the installation of the selected Tivoli Workload Scheduler component, follow the instructions described in the following sections.

To access information about product installation prerequisites, click the different options in the left frame of the launchpad.

**Installation wizard**

Install Tivoli Workload Scheduler master domain managers, backup masters, agents, connectors, and the Dynamic Workload Console by launching the individual setup files for each supported platform.

You can use the installation wizard in interactive or silent mode. In interactive mode, the wizard guides you through the installation steps. In silent mode, a response file provides the relevant information to the installation process, which is run in background.

This method of installation uses a Java Virtual Machine, and therefore has specific system requirements. See "Checking prerequisites (UNIX and Linux)" on page 24 for details about installation requirements.

**Silent mode**

Customize a response file by adding all the configuration settings to be used during installation. Then, from the command line, run the setup command. Using this method you can run the installation unattended and in the background. For more information, see "Performing a silent installation" on page 84.

**The twsinst script for agents**

Only use twsinst to install Tivoli Workload Scheduler agents if you are not running a Java Virtual Machine (JVM) on the workstation. Otherwise, perform a silent installation instead. See "Performing a silent installation" on page 84.

After you use the twsinst script to upgrade an instance of Tivoli Workload Scheduler to version 8.6, whose previous version was installed using the installation wizard, you can only use the twsinst method to upgrade that instance of Tivoli Workload Scheduler again. However, you can uninstall that Tivoli Workload Scheduler instance using the wizard.

For information about twsinst, see "Installing agents using twsinst" on page 88.

**Software Distribution software package blocks (SPBs)**

Install agents using the Software Distribution component of IBM Tivoli Configuration Manager, versions 4.1, 4.2, 4.2.1, 4.2.2, or 4.2.3 by distributing software package blocks. See "Installing agents using Software Distribution" on page 96.

Use Software Distribution to install Tivoli Workload Scheduler agents only if you do not run a JVM on the workstation. If this is not your situation, you might choose to perform a silent installation instead. See "Performing a silent installation" on page 84.
Installation log files

The type of log files you find on your system depends on the type of installation you performed. This section describes the logs associated with the different installations.

For more information about log files, see the Administration Guide.

InstallShield wizard installation and uninstallation log files

You can check the following log files for information about the installation. Details of the installation process are kept in log files on the local workstation in the following directories:

Tivoli Workload Scheduler

Windows operating systems:
%Temp%\TWA\tws86
where
%Temp% is the windows system environment variable.

UNIX and Linux operating systems:
/tmp/TWA/tws86

Table 4 lists the InstallShield wizard log files.

<table>
<thead>
<tr>
<th>Log file name</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>twsstatus.log</td>
<td>Tivoli Workload Scheduler installation status log file. It reports if the installation completed successfully or with errors. In case of errors it indicates if the error is due to an incorrect field value or to a failed step.</td>
</tr>
<tr>
<td>twsismp.log</td>
<td>Tivoli Workload Scheduler installation trace file</td>
</tr>
<tr>
<td>summary.log</td>
<td>Tivoli Workload Scheduler installation log file</td>
</tr>
</tbody>
</table>

For multiple installations on the same workstation, the log header and footer indicate the user ID (<TWS_user>) for which the installation was performed. Most log files are overwritten if there are multiple installations on the same workstation. The exceptions are the following files, which are not overwritten but appended:
- twsismp.log
- summary.log

Note: If you are running a silent installation and the response file you are using does not have the correct syntax, the installation fails without producing a log file.

The twsinst log files

The twsinst log file is as follows:

<tempDir>/twsinst_<operating_system>_<TWS_user>^8.6.0.00.log, where:

<tempDir>
The user temporary directory:
UNIX /tmp and /tmp/TWA/tws86
Windows operating system
%Temp%\TWA\tws86
The operating system.

The name of the user for which Tivoli Workload Scheduler was installed (the name you supplied during installation)

Software package block log files
The IBM Tivoli Configuration Manager software package block log files are as follows:

- `<tempDir>/TWA/tws86/FP_TWS_<operating_system>_<TWS_user>^8.6.0.00.log` (agent SPB log file)
- `<tempDir>/TWA/tws86/TWS_LP_<TWS_user>^8.6.0.00.log` (agent NLS SPB log file)

where:

- `<tempDir>`
  - The user temporary directory:
    - **Windows operating systems**
      - The default value is: `%Temp%\TWA\tws86`
    - **UNIX and Linux operating systems**
      - `/tmp/TWA/tws86`

- `<operating_system>`
  - The operating system.

- `<TWS_user>`
  - The name of the user for which Tivoli Workload Scheduler was installed (the name you supplied during installation)

WebSphere Application Server installation log files
The application server installation has no log. However, if you update the application server, for example during the application of a Tivoli Workload Scheduler fix pack, a log is created which gives information about the update. The log can be found in the directory `<TWS_home>/eWAS/logs/update`, where you can find a directory that identifies the fix pack that has been installed, for example: `7.0.0-WS-WASEmbedded-AixPPC64-FP000027.install`, which contains a log file called `updateLog.txt`.

The log for the startup of the application server can be found at: `<TWS_home>/eWAS/profiles/TIPProfile/logs/<SERVERNAME>/startServer.log`

where `<SERVERNAME>` is:
- `server1` in the first instance of Tivoli Workload Automation
- `server2, server3...` for subsequent instances of Tivoli Workload Automation

**Note:** In the case of an upgrade, `<SERVERNAME>` is the name of the previous server, for example `server1`.

DB2 installation log files
For information about DB2 installation log files, see the DB2 documentation.
Tivoli Workload Scheduler user

On UNIX and Linux operating systems, regardless of the method of installation you choose, the Tivoli Workload Scheduler user must be created manually before running the installation. Use the appropriate UNIX and Linux operating system commands to create the user.

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

Windows users domain rights and structure

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

Do not install two agents on the same system, one on the local system and one defined on the domain with the same name, using the same user.

The Windows user performing the installation must:
• For a local TWS user, be a member of the local administrative group
• For a domain TWS user, be a member of the domain administrative groups in the domain controller

All Windows 2003 Tivoli Workload Scheduler users need the following user rights. They can be granted locally. Domain level policies always override local policies, so you might be required to grant the following rights 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
  – As a service
• Replace process level token

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

Considerations for Windows domain controllers running Microsoft Active Directory

If you want to install Tivoli Workload Scheduler fault-tolerant agents on workstations where users who run jobs are domain users and the domain controller is the running Microsoft Active Directory, decide how to install the agents and configure the domain so that the jobmon process can obtain the correct information to let the users run jobs.

Before running a job, jobmon must retrieve 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.
To resolve this issue, 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 the `<TWS_user>` for each. In this case, only the 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 the domain users can run the jobs submitted by jobmon.

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

Set up the `<TWS_user>` as a Windows domain user and install the instance of Tivoli Workload Scheduler 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 the local or 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 was in a Windows domain which does not support Active Directory. In this case, all the local and the domain users can run the jobs submitted by jobmon. In addition, the domain controller does not prevent any local or domain user from running other processes that are not controlled by Tivoli Workload Scheduler.

### Checking environment settings for Windows Vista users

Before you install Tivoli Workload Scheduler on a Windows Vista workstation that does not belong to a Windows domain, make sure that the workstation name and the domain name are both registered in uppercase in the Windows environment settings. When the workstation is not in a Windows domain, the COMPUTERNAME and USERDOMAIN values are identical, but on Vista the USERDOMAIN value is sometimes in lowercase even if the COMPUTERNAME is in uppercase.

To resolve this issue, perform the following actions:

1. Open a DOS command prompt shell.
2. Run the `set` command to display the Windows environment settings.
3. Check that the USERDOMAIN value is in uppercase. If this is not the case, follow this workaround to correct it:
4. Run the `set` command to change the value of COMPUTERNAME to a temporary host name of your choice:
   ```plaintext```
   set /p COMPUTERNAME=MYTEMPHOST
   ```plaintext```
5. Restart the system.
6. Run the `set` command again as in step 4 replacing the temporary host name with the original one.
7. Restart the system.
8. Check that the USERDOMAIN value is now in uppercase.
Windows services

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

- Tivoli Workload Scheduler (for <TWS_user>)
- Tivoli Netman (for <TWS_user>)
- Tivoli Token Service (for <TWS_user>) - includes the In-Flight Tracing facility service
- Tivoli Workload Scheduler SSM Agent (for <TWS_user>)
- WebSphere Application Server (for <TWS_user>)
- IBM Common Platform Agent: tws_cpa_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 following installation, you must use the Services applet in the Control Panel to assign the new password for the Tivoli Token Service and Tivoli Workload Scheduler (for <TWS_user>). For more information, see the section on changing the password of the TWS_User in Administration Guide.

Creating or upgrading the Tivoli Workload Scheduler database tables before installing or upgrading

This procedure is not mandatory for the installation of the product. The database administrator must run the procedure to create or upgrade the product database tables before the installation of the product only when the IT administrator who installs the product does not know the database administrator user ID and password. Otherwise, the IT administrator provides the database administrator user ID and password during the installation and the tables are automatically created and upgraded during the installation or the upgrade of the product.

Using this procedure, the database administrator creates or upgrades the database tables before the IT administrator installs or upgrades the product with a user different from the database administrator user. The procedure ensures that only the database administrator manages all the confidential information related to the database, such as the administrator user ID and password, and the IT administrator can install or upgrade the product without knowing any confidential database information.

This chapter describes the procedure to follow if you want to:

- Create and upgrade the Tivoli Workload Scheduler and the dynamic workload broker database tables before installing or upgrading the product if you are using DB2. See “Creating or upgrading the database tables if you are using DB2” on page 37.
- Create and upgrade the Tivoli Workload Scheduler and the dynamic workload broker database tables before installing or upgrading the product if you are using Oracle. See “Creating or upgrading the database tables if you are using Oracle” on page 47.

The IT administrator can perform:

- The installation, specifying as database administrator user name the user to be granted access, by the administrator of the DB2 server, to the Tivoli Workload Scheduler database.
The upgrade, by using another user that has the same permissions as the user that installed the product.

Creating or upgrading the database tables if you are using DB2

To create or upgrade the Tivoli Workload Scheduler and the dynamic workload broker database tables if you are using DB2, run the following procedures:

1. Customize the properties file. See “Customizing the properties file for DB2.”
2. Generate the SQL files. See “Generating the SQL files for DB2” on page 39.
3. Run the SQL files to create the SQL tables. See “Running the SQL files to create or upgrade the SQL tables for DB2” on page 40.

Customizing the properties file for DB2

To customize the properties files, perform the following steps:

1. From the installation DVD or from the elmage containing the master domain manager or the dynamic domain manager, open the following properties file:

   On Windows operating systems:
   `<images_dir>\RESPONSEFILES\customizeDB2SQL.properties`

   On UNIX and Linux systems:
   `<images_dir>/RESPONSEFILES/customizeDB2SQL.properties`

   where: `images_dir` specifies the directory where you extract the product image.

2. Customize the SQL properties with the values appropriate for your needs:

   **TWSTEMPDIR**
   The directory where you want to store the SQL scripts to create the database tables. The default value is:
   
   On Windows operating system:
   `C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\TWA\tws86`

   On UNIX and Linux systems:
   `/tmp/TWA/tws86`

   **DB_USER**
   If you are creating the database tables before installing the product:
   Specify the user to be granted access by the administrator of the DB2 server to access the Tivoli Workload Scheduler database.

   When the IT administrator installs the product, he must specify this value in the **DB2 server administrator user** field.

   On Windows operating system
   The default value is **db2admin**.

   On UNIX and Linux operating systems
   The default value is **db2inst1**.

   On UNIX, verify that you can switch to this user and that it can load the DB2 environment.

   If you are upgrading the database tables before upgrading the product:
   Specify the user that you used when you installed the version of the product you are upgrading. When you later upgrade the
product, you can specify a user different from the one you specified in the DB_USER field, but it must have database access permissions.

**TWS_USER**
Specify the Tivoli Workload Scheduler user name.

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

When the IT administrator installs the product, he must specify this value in the **User_name** field.

**TWS_DB**
The name of the DB2 database. The maximum length is five characters. The default value is **TWS**. If you are creating the SQL tables for a:

- **Master domain manager**
  Provide the name of a database that is not used by a dynamic domain manager.

- **Dynamic domain manager**
  Provide the name of a database that is not used by a master domain manager.

When the IT administrator installs the product, he must specify this value in the **Database name** field.

**TWS_TS_NAME**
The name of the DB2 instance table space. This table space is used to store scheduling objects and event rules. For information about DB2 table spaces, see the DB2 documentation. The default table space name is **TWS_DATA**.

When the IT administrator installs the product, he must specify this value in the **Tablespace name** field.

**TWS_DATA_TS_PATH**
The relative path of the DB2 table space. The path can be a relative or a fully qualified path. When the table space path is a fully qualified path, the DB2 administrator user must have complete access rights to the directory where the table space is installed. The default table space path name is **TWS_DATA**. For UNIX and Linux operating systems, make sure that the DB2 administrator has write access to the directory above the table space directory. For more information, see "DB2 tablespace relative paths" on page 312.

When the IT administrator installs the product, he must specify this value in the **Tablespace path** field.

**TWS_LOG_TS_NAME**
Specify the name of the DB2 table space where Tivoli Workload Scheduler event logs are to be stored. These logs include data about event rule instances, triggered actions, and operator messages that are displayed by the Dynamic Workload Console. Data from the logs can be used to create reports. You can view report data by using the Dynamic Workload Console. The default name is **TWS_LOG**.

When the IT administrator installs the product, he must specify this value in the **Report tablespace name** field.
TWS_LOG_TS_PATH
Specify the path of the DB2 table space where Tivoli Workload Scheduler event logs are to be stored. The default path is TWS_LOG. The path can be a relative or a fully qualified path. When the table space path is a fully qualified path the DB2 administrator user must have complete access rights to the directory where the table space is installed. For more information, see “DB2 tablespace relative paths” on page 312.

Note: The report table space path cannot be the same as the table space path.

When the IT administrator installs the product, he must specify this value in the **Report tablespace path** field.

COMPANY_NAME
The name of the company. You can use spaces and the maximum field length is 40 characters. The default is MYCOMPANY.

When the IT administrator installs the product, he must specify this value in the **Company** field.

EIF_PORT
The port used by the event management processor to receive events. The default value is 31131. The valid range is from 1 to 65535.

When the IT administrator installs the product, he must specify this value in the **Event Processor** field.

HOST_NAME
The fully qualified host name or IP address on which the dynamic domain manager is contacted by the dynamic agent.

When the IT administrator installs the product, he must specify this value in the dynamic agent configuration information **Host name or IP address** field.

WAS_SEC_PORT
The HTTPS port of the dynamic workload broker. The dynamic agent uses it to connect to the dynamic workload broker. The default value is 31116. If you leave the field blank, it defaults to 0. The valid range is from 1 to 65535.

When the IT administrator installs the product, he must specify this value in the **Dynamic workload broker HTTPS port number** field.

**Generating the SQL files for DB2**
To generate the SQL files you use the `customizeSQL` script. It is located:

On Windows operating system

```
<images_dir>/TWS/<operating_system>/tws_tools/customizeSQL.bat
```

On UNIX and Linux operating systems

```
<images_dir>/TWS/<operating_system>/tws_tools/customizeSQL.sh
```

where:

*images_dir*
The directory where you stored the product images. If you want to use the image, download the one containing the master domain manager.

*operating_system*
The operating system you are working on.
To show command usage, run:
customizeSQL -usage

The script has the following syntax:

**On Windows operating system**
customizeSQL -javaHome <javahome_dir> -propertyFile <property_file_path>

**On UNIX and Linux operating systems**
customizeSQL -imagesDir <images_dir> -javaHome <javahome_dir>
   -propertyFile <property_file_path>

where:

*images_dir*
   UNIX only. The directory where the Tivoli Workload Scheduler setup.bin command is located.

*javahome_dir*
   The directory where the Java runtime environment is located.

*property_file_path*
   The directory where the property file is located.

The following example shows the command for UNIX if you stored the images in the /opt/TWSImages/TWS/AIX directory and the properties file in the /tmp/ directory:
customizeSQL -imagesDir /opt/TWSImages/TWS/AIX -javaHome /usr -propertyFile /tmp/customizeDB2SQL.properties

The SQL files are created in the directory you specified in the **TWSTEMPDIR** property.

The following example shows the command for Windows if you stored the Java runtime in the C:\Program Files\Java\jre6 and the properties file in the C:\Temp\ directory:
customizeSQL -javaHome "C:\Program Files\Java\jre6" -propertyFile "C:\Temp\customizeDb2Sql.properties"

The SQL files are created in the directory you specified in the **TWSTEMPDIR** property.

**Running the SQL files to create or upgrade the SQL tables for DB2**

This section describes the command you must run to create or upgrade the SQL tables. The commands you must run depends if you are:

**Creating the SQL tables before installing:**
- A master domain manager. See "Before installing a master domain manager" on page 41.
- A dynamic domain manager. See "Before installing a dynamic domain manager" on page 42.

**Upgrading the SQL tables before upgrading the product**
- From version 8.3 up to version 8.5. See "Before upgrading Tivoli Workload Scheduler from V8.3 and later to V8.5" on page 43.
- From version 8.5.1 up to version 8.5.1.1. See "Before upgrading Tivoli Workload Scheduler from V8.5.1 to V8.5.1.1" on page 45.
These scripts must be run by the database administrator by using the `db2inst1` administrator user or the user that the database administrator granted access to the Tivoli Workload Scheduler database.

**Before installing a master domain manager:** If you are creating the SQL tables before installing a master domain manager, perform the following steps:

1. Create the Tivoli Workload Scheduler database, by running the following command:
   ```
   db2 -v -t -f <sql_directory>/sql/create_database.sql
   ```
   where:
   - `sql_directory` is the directory where you stored the SQL you created in "Generating the SQL files for DB2" on page 39.

2. Connect the user to the Tivoli Workload Scheduler database, by running the following command:
   ```
   db2 connect to TWS_DB user ADMIN_USER "ADMIN_PW"
   ```
   where:
   - `TWS_DB` is the value you specified in the `TWS_DB` property.
   - `ADMIN_USER` is the DB2 database administrator.
   - `ADMIN_PW` is the password of the DB2 database administrator.

3. Create the Tivoli Workload Scheduler table spaces, by running the following command:
   ```
   db2 -v -t -f <sql_directory>/sql/create_tablespaces.sql
   ```

4. Create the Tivoli Workload Scheduler event rule tables, by running the following command:
   ```
   db2 -v -t -f <sql_directory>/sql/create_rule_tables.sql
   ```

5. Create the Tivoli Workload Scheduler log tables, by running the following command:
   ```
   db2 -v -t -f <sql_directory>/sql/create_log_tables.sql
   ```

6. Create the Tivoli Workload Scheduler tables, by running the following command:
   ```
   db2 -v -t -f <sql_directory>/sql/create_tables.sql
   ```

7. Set the Tivoli Workload Scheduler event rule constraints, by running the following command:
   ```
   db2 -v -t -f <sql_directory>/sql/create_rule_constraints.sql
   ```

8. Set the Tivoli Workload Scheduler log constraints, by running the following command:
   ```
   db2 -v -t -f <sql_directory>/sql/create_log_constraints.sql
   ```

9. Set the Tivoli Workload Scheduler constraints, by running the following command:
   ```
   db2 -v -t -f <sql_directory>/sql/create_constraints.sql
   ```

10. Populate the Tivoli Workload Scheduler tables, by running the following command:
    ```
        db2 -v -t -f <sql_directory>/sql/populate_tables.sql
    ```

11. Create the Tivoli Workload Scheduler event rule indexes, by running the following command:
db2 -v -t -f $sql_directory/sql/create_rule_indexes.sql

12. Create the Tivoli Workload Scheduler log indexes, by running the following command:
   db2 -v -t -f $sql_directory/sql/create_log_indexes.sql

13. Create the Tivoli Workload Scheduler indexes, by running the following command:
   db2 -v -t -f $sql_directory/sql/create_indexes.sql

14. Create the Tivoli Workload Scheduler event rule views, by running the following command:
   db2 -v -t -f $sql_directory/sql/create_rule_views.sql

15. Create the Tivoli Workload Scheduler log views, by running the following command:
   db2 -v -t -f $sql_directory/sql/create_log_views.sql

16. Create the Tivoli Workload Scheduler views, by running the following command:
   db2 -v -t -f $sql_directory/sql/create_views.sql

17. Assign permissions to the Tivoli Workload Scheduler user to access Tivoli Workload Scheduler tables and views, by running the following command:
   db2 -v -t -f $sql_directory/sql/grant_twsuser.sql

18. Grant permissions to the database user, by running the following command:
   db2 grant DBADM on database to user DB_USER

19. Create the job repository, by running the following command:
   db2 -v -t -f $sql_directory/DWB/sql/create_job_repository.sql

20. Create the allocation repository, by running the following command:
   db2 -v -t -f $sql_directory/DWB/sql/create_allocation_repository.sql

21. Create the resource repository, by running the following command:
   db2 -v -t -f $sql_directory/DWB/sql/create_resource_repository.sql

22. Create the server table, by running the following command:
   db2 -v -t -f $sql_directory/DWB/sql/create_server.sql

23. Catalog the Tivoli Workload Scheduler node, by running the following command:
   db2 catalog local node TWS_DB_ND instance DB2_instance_name

24. Catalog the local node, by running the following command:
   db2 catalog tcpip node LBNODE remote 127.0.0.1 server DB2_instance_port

25. Catalog the alias of the Tivoli Workload Scheduler database, by running the following command:
   db2 catalog db TW5_DB as TW5_DB_DB at node LBNODE

**Before installing a dynamic domain manager:** If you are creating the SQL tables before installing a dynamic domain manager, perform the following steps:

1. Create the Tivoli Workload Scheduler database, by running the following command:
   db2 -v -t -f $sql_directory/sql/create_database.sql

where:

$sql_directory$ is the directory where you stored the SQL you created in

"Generating the SQL files for DB2" on page 39
2. Connect the user to the Tivoli Workload Scheduler database, by running the following command:
   
   ```
   db2 connect to TWS_DB user ADMIN_USER "ADMIN_PW"
   ```

   where:
   
   **TWS_DB**
   
   Is the value you specified in the TWS_DB property.
   
   **ADMIN_USER**
   
   Is the DB2 database administrator.
   
   **ADMIN_PW**
   
   Is the password of the DB2 database administrator.

3. Create the Tivoli Workload Scheduler table spaces, by running the following command:
   
   ```
   db2 -v -t -f sql_directory\sql\create_tablespaces.sql
   ```

4. Create the job repository, by running the following command:
   
   ```
   db2 -v -t -f sql_directory\DWB\sql\create_job_repository.sql
   ```

5. Create the allocation repository, by running the following command:
   
   ```
   db2 -v -t -f sql_directory\DWB\sql\create_allocation_repository.sql
   ```

6. Create the resource repository, by running the following command:
   
   ```
   db2 -v -t -f sql_directory\DWB\sql\create_resource_repository.sql
   ```

7. Create the server table, by running the following command:
   
   ```
   db2 -v -t -f sql_directory\DWB\sql\create_server.sql
   ```

8. Grant permissions to the database user, by running the following command:
   
   ```
   db2 grant DBADM on database to user DB_USER
   ```

9. Catalog the alias of the Tivoli Workload Scheduler database, by running the following command:
   
   ```
   db2 catalog db TWS_DB as TWS_DB_DB
   at node LBNODE
   ```

10. Catalog the local node, by running the following command:
    
    ```
    db2 catalog tcpip node LBNODE remote 127.0.0.1 server DB2_instance_port
    ```

**Before upgrading Tivoli Workload Scheduler from V8.3 and later to V8.5:** If you want to upgrade the SQL tables before upgrading Tivoli Workload Scheduler from V8.3 and later to V8.5, perform the following steps:

1. Connect the user to the Tivoli Workload Scheduler database, by running the following command:
   
   ```
   db2 connect to TWS_DB user ADMIN_USER using "ADMIN_PW"
   ```

   where:
   
   **TWS_DB**
   
   Is the value you specified in the TWS_DB property.
   
   **ADMIN_USER**
   
   Is the DB2 database administrator.
   
   **ADMIN_PW**
   
   Is the password of the DB2 database administrator.

2. Grant permissions to the database user, by running the following command:
   
   ```
   db2 grant DBADM on database to user DB_UPGRADE_USER
   ```
where DB_UPGRADE_USER is the name of the user that the IT administrator specifies in the DB2 Server administrator user field later when upgrading the product. You must assign SYSMON authority to the DB_UPGRADE_USER.

4. Verify the version of your database, by running the `select` command.
   The command must be run by a user that has DBADM authority or at least DATAACCESS authority.
   Run the command by using the following syntax:
   ```sql
   select MPR_DESCRIPTION, MPR_VALUE from MDL.MPR_MODEL_PROPERTIES
   where MPR_NAME='DATABASE_VERSION'
   ```
   If `DATABASE_VERSION` returns one of the following values:
   - **8.3.0.00** You have installed version 8.3.
   - **8.3.0.01** You have installed version 8.3.0.1.
   - **8.4.0.00** You have installed version 8.4.
   - **8.4.0.01** You have installed version 8.4.0.1.
   - **8.5.0.00** You have installed version 8.5.

   run the procedure described in Step 5
   If `DATABASE_VERSION` returns one of the following values:
   - **8.5.1.00** You have installed version 8.5.1.
   - **8.5.1.01** You have installed version 8.5.1.1.

   run the procedure described in "Before upgrading Tivoli Workload Scheduler from V8.5.1 to V8.5.1.1" on page 45.

5. Run the following commands in sequence starting with the command for your specific `DATABASE_VERSION` value.
   For example, if `DATABASE_VERSION` is equal to:
   - **8.3.0.00** Run all the commands listed in To upgrade the Tivoli Workload Scheduler database tables.
   - **8.4.0.00** Run the following commands:
     a. 4c on page 45
     b. 4d on page 45
     c. 4e on page 45
     d. 4f on page 45
     e. 4g on page 45
     f. 4h on page 45
   - **8.5.0.00** To upgrade the Tivoli Workload Scheduler database tables, run the following commands:
     a. 4e on page 45
     b. 4f on page 45
     c. 4g on page 45
     d. 4h on page 45
To upgrade the dynamic workload broker database tables, run the commands listed in the upgraded dynamic workload broker database tables.

To upgrade the Tivoli Workload Scheduler database tables:

a. db2 -v -t -f <sql_directory>/sql/upgrade_83000_83001.sql

b. db2 -v -t -f <sql_directory>/sql/upgrade_83001_84000.sql

c. db2 -v -t -f <sql_directory>/sql/upgrade_84000_84001.sql

d. db2 -v -t -f <sql_directory>/sql/upgrade_84001_85000.sql

e. db2 -v -t -f <sql_directory>/sql/upgrade_85000_85100.sql

f. db2 -v -t -f <sql_directory>/sql/upgrade_85100_85101.sql

g. db2 -v -t -f <sql_directory>/sql/upgrade_85101_86000.sql

h. db2 -x 'SELECT COUNT(*) FROM MDL.JSF_JS_INST_FORECAST'
   • If the command returns the following error message:
   SQL0204N "MDL.JSF_JS_INST_FORECAST" name is an undefined name. SQLSTATE=42704

   run the following command:
   db2 -v -t -f <sql_directory>/sql/upgrade_84001_84005.sql

To upgrade the dynamic workload broker database tables:

a. Create the job repository, by running the following command:
   db2 -v -t -f <sql_directory>/DWB/sql/create_job_repository.sql

b. Create the allocation repository, by running the following command:
   db2 -v -t -f <sql_directory>/DWB/sql/create_allocation_repository.sql

c. Create the resource repository, by running the following command:
   db2 -v -t -f <sql_directory>/DWB/sql/create_resource_repository.sql

d. Create the server table, by running the following command:
   db2 -v -t -f <sql_directory>/DWB/sql/create_server.sql

Before upgrading Tivoli Workload Scheduler from V8.5.1 to V8.5.1.1: If you want to upgrade the SQL tables before upgrading Tivoli Workload Scheduler from version 8.5.1 to version 8.5.1.1, perform the following steps:

1. Connect the user to the Tivoli Workload Scheduler database, by running the following command:
   db2 connect to TWS_DB user ADMIN_USER using "$ADMIN_PW"

   where:
   TWS_DB
   Is the value you specified in the TWS_DB property.
**ADMIN_USER**
Is the DB2 database administrator.

**ADMIN_PW**
Is the password of the DB2 database administrator.

2. Grant permissions to the database user, by running the following command:

```sql
db2 grant DBADM on database to user DB_UPGRADE_USER
```

where **DB_UPGRADE_USER** is the name of the user that the IT administrator specifies in the **DB2 Server administrator user** field when later upgrading the product. You must assign SYSMON authority to the **DB_UPGRADE_USER**.

3. Verify the version of your database, by running the **select** command.

The command must be run by a user that has DBADM authority or at least DATAACCESS authority. Run the command by using the following syntax:

```sql
select MPR_DESCRIPTION, MPR_VALUE from MDL.MPR_MODEL_PROPERTIES
where MPR_NAME='DATABASE_VERSION'
```

If **DATABASE_VERSION** returns one of the following values:

- **8.3.0.00**
  You have installed version 8.3.
- **8.3.0.01**
  You have installed version 8.3.0.1.
- **8.4.0.00**
  You have installed version 8.4.
- **8.4.0.01**
  You have installed version 8.4.0.1.
- **8.5.0.00**
  You have installed version 8.5.

run the procedure described in "Before upgrading Tivoli Workload Scheduler from V8.3 and later to V8.5" on page 43.

If **DATABASE_VERSION** returns one of the following values:

- **8.5.1.00**
  You have installed version 8.5.1.
- **8.5.1.01**
  You have installed version 8.5.1.1.

run the procedure described in Step 4.

4. If **DATABASE_VERSION** returns a value from 8.5.1.00 to 8.5.1.01, run the following commands in sequence:

**To upgrade the Tivoli Workload Scheduler database tables:**

a.

```bash
db2 -v -t -f <sql_directory>/sql/upgrade_85100_85101.sql
```

b.

```bash
db2 -v -t -f <sql_directory>/sql/upgrade_85101_86000.sql
```

c.

```bash
db2 -x 'SELECT COUNT (*) FROM MDL.JSF_JS_INST_FORECAST'
```

- If the command returns the following error message:

  SQL0204N "MDL.JSF_JS_INST_FORECAST" name is an undefined name. SQLSTATE=42704

  run the following command:

```bash
db2 -v -t -f <sql_directory>/sql/upgrade_84001_84005.sql
```
To upgrade the dynamic workload broker database tables:

    db2 -v -t -f <sql_directory>/DWB/sql/upgrade_85100_86000.sql

Creating or upgrading the database tables if you are using Oracle

To create or upgrade the Tivoli Workload Scheduler and the dynamic workload broker database tables if you are using Oracle, run the following procedures:
1. Customize the properties file. See “Customizing the properties file for Oracle.”
2. Generate the SQL files. See “Generating the SQL files for Oracle” on page 48.
3. Run the SQL files to create the SQL tables. See “Running the SQL files to create or upgrade the SQL tables for Oracle” on page 49.

Customizing the properties file for Oracle

To customize the properties files, perform the following steps:
1. From the installation DVD or from the image containing the master domain manager or the dynamic domain manager, open the following properties file:
   
   **On Windows operating systems:**
   
   `<images_dir>\RESPONSEFILES\customizeORACLESQL.properties`

   **On UNIX and Linux systems:**
   
   `<images_dir>/RESPONSEFILES/customizeORACLESQL.properties`

   where: `<images_dir>` specifies the directory where you extract the product image.
2. Customize the SQL properties with the values appropriate for your needs:
   
   **TWSTEMPDIR**
   
   The directory where you want to store the SQL scripts to create the database tables. The default value is:

   **On Windows operating systems:**
   
   `C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\TWA\tws86`

   **On UNIX and Linux systems:**
   
   `/tmp/TWA/tws86`

   **MDL_USER**
   
   The database administrator user name (such as `SYSTEM`) that access the Tivoli Workload Scheduler and the dynamic workload broker database. It is a user that must be created on Oracle; it is not an operating system user.

   When the IT administrator installs the product, he must specify this value in the **Oracle administrator user** field.

   **TWS_PASSWORD**
   
   The password of the `MDL_USER`.

   When the IT administrator installs the product, he must specify this value in the **Oracle administrator user password** field.

   **TWS_USER**
   
   Specify the Tivoli Workload Scheduler user name.

   It can contain alphanumeric, dash (`-`), and underscore (`_`) characters; it cannot contain national characters. The first character of the user name must be a letter.
When the IT administrator installs the product, he must specify this value in the **User_name** field.

**TWS_TS_NAME**
The name that identifies the Tivoli Workload Scheduler data table space. The default for this field is **USERS**.

When the IT administrator installs the product, he must specify this value in the **Tivoli Workload Scheduler data tablespace** field.

**TWS_LOG_TS_NAME**
The name that identifies the Tivoli Workload Scheduler table space where report data is to be stored. You can view the report data by using the Dynamic Workload Console. The default value for this field is **USERS**.

When the IT administrator installs the product, he must specify this value in the **Tivoli Workload Scheduler reports tablespace** field.

**TWS_TS_TEMP_NAME**
The name that identifies the Tivoli Workload Scheduler temporary table space. The default value for this field is **TEMP**.

When the IT administrator installs the product, he must specify this value in the **Tivoli Workload Scheduler temporary tablespace** field.

**COMPANY_NAME**
The name of the company. You can use spaces and the maximum field length is 40 characters. The default is **MYCOMPANY**.

When the IT administrator installs the product, he must specify this value in the **Company** field.

**EIF_PORT**
The port used by the event management processor to receive events. The valid range is from 1 to 65535. The default value is **31131**.

When the IT administrator installs the product, he must specify this value in the **Event Processor** field.

**HOST_NAME**
The fully qualified host name or IP address on which the dynamic domain manager is contacted by the dynamic agent.

When the IT administrator installs the product, he must specify this value in the dynamic agent configuration information **Host name or IP address** field.

**WAS_SEC_PORT**
The HTTPS port of the dynamic workload broker. The dynamic agent uses it to connect to the dynamic workload broker. The valid range is from 1 to 65535. The default value is **31116**. If you leave the field blank, it defaults to **0**.

When the IT administrator installs the product, he must specify this value in the **Dynamic workload broker HTTPS port number** field.

---

**Generating the SQL files for Oracle**

To generate the SQL tables, run the `customizesql` script as described in “Generating the SQL files for DB2” on page 39.
Running the SQL files to create or upgrade the SQL tables for Oracle

This section describes the commands you must run to create or upgrade the SQL tables. The commands you must run depends if you are:

Creating the SQL tables before installing:
- A master domain manager. See “Before installing a master domain manager.”
- A dynamic domain manager. See “Before installing a dynamic domain manager” on page 52.

Upgrading the SQL tables before upgrading the product
- From version 8.3 and later to version 8.5. See “Before upgrading Tivoli Workload Scheduler from V8.3 and later to V8.5” on page 53.
- From version 8.5.1 up to version 8.5.1.1. See “Before upgrading from V8.5.1 to V8.5.1.1” on page 57.

These scripts must be run by the database administrator by using the value you specified in MDL_USER or the user that the database administrator granted to access the Tivoli Workload Scheduler database.

Before installing a master domain manager: If you are creating the SQL tables before installing a master domain manager, perform the following steps:

1. Create the Tivoli Workload Scheduler users, by running the following command:

   sqlplus -L -S <ORACLE_ADMIN>/<ORACLE_ADMIN_PW>
   @<ORACLE_NET_SERVICE_NAME>
   @<sql_directory>/sql/create_users.sql

   where:

   ORACLE_ADMIN
   Is the Oracle database administrator.

   ORACLE_ADMIN_PW
   Is the password of the ORACLE_ADMIN.

Net service name

The name used by clients to identify an Oracle Net server and the specific system identifier or database for the Oracle Net connection. A net service name is mapped to a port number and protocol. It is also known as a connect string, database alias, host string, or service name.

If your Oracle database is:
- Installed on the same system where you are installing your master domain manager or a backup master, the net service name is the name of your Oracle database.
- Installed on the same system where you are installing your dynamic domain manager or a backup dynamic domain manager, the net service name is the name of your Oracle database.
- Not installed on the system where you are installing your master domain manager or a backup master, the net service name is the alias configured for the connection to the remote database.
Not installed on the system where you are installing your dynamic domain manager or a backup dynamic domain manager, the net service name is the alias configured for the connection to the remote database.

When the IT administrator installs the product he must specify this value in the **Net service name** field.

**sql_directory**

The directory where you stored the SQL you created in “Generating the SQL files for DB2” on page 39.

2. Create the Tivoli Workload Scheduler event rule tables, by running the following command:

```
sqlplus -L -S <TWS_MDL_USER>/<TWS_MDL_PW>
@<ORACLE_NET_SERVICE_NAME>
@<sql_directory>/sql/create_rule_tables.sql
```

where:

- **TWS_MDL_USER**
  - Is the user you specified in the **MDL_USER** field of the property file.

- **TWS_MDL_PW**
  - Is the password of the **MDL_USER**.

3. Create the Tivoli Workload Scheduler log tables, by running the following command:

   **If you did not use the ORACLE partitioning feature**

   ```
   sqlplus -L -S <TWS_MDL_USER>/<TWS_MDL_PW>
   @<ORACLE_NET_SERVICE_NAME>
   @<sql_directory>/sql/stded/create_log_tables.sql
   ```

   **If you used the ORACLE partitioning feature**

   ```
   sqlplus -L -S <TWS_MDL_USER>/<TWS_MDL_PW>
   @<ORACLE_NET_SERVICE_NAME>
   @<sql_directory>/sql/create_log_tables.sql
   ```

4. Create the Tivoli Workload Scheduler tables, by running the following command:

```
sqlplus -L -S <TWS_MDL_USER>/<TWS_MDL_PW>
@<ORACLE_NET_SERVICE_NAME>
@<sql_directory>/sql/create_tables.sql
```

5. Set the Tivoli Workload Scheduler event rule constraints, by running the following command:

```
sqlplus -L -S <TWS_MDL_USER>/<TWS_MDL_PW>
@<ORACLE_NET_SERVICE_NAME>
@<sql_directory>/sql/create_rule_constraints.sql
```

6. Set the Tivoli Workload Scheduler log constraints, by running the following command:

```
sqlplus -L -S <TWS_MDL_USER>/<TWS_MDL_PW>
@<ORACLE_NET_SERVICE_NAME>
@<sql_directory>/sql/create_log_constraints.sql
```

7. Set the Tivoli Workload Scheduler constraints, by running the following command:

```
sqlplus -L -S <TWS_MDL_USER>/<TWS_MDL_PW>
@<ORACLE_NET_SERVICE_NAME>
@<sql_directory>/sql/create_constraints.sql
```

8. Populate the Tivoli Workload Scheduler tables, by running the following command:
9. Create the Tivoli Workload Scheduler event rule indexes, by running the following command:

```sql
sqlplus -L -S <TWS_MDL_USER>/TWS_MDL_PW
@<ORACLE_NET_SERVICE_NAME>
@<sql_directory>/sql/populate_tables.sql
```

10. Create the Tivoli Workload Scheduler log indexes, by running the following command:

**If you did not use the ORACLE partitioning feature**

```sql
sqlplus -L -S <TWS_MDL_USER>/TWS_MDL_PW
@<ORACLE_NET_SERVICE_NAME>
@<sql_directory>/sql/stded/create_log_indexes.sql
```

**If you used the ORACLE partitioning feature**

```sql
sqlplus -L -S <TWS_MDL_USER>/TWS_MDL_PW
@<ORACLE_NET_SERVICE_NAME>
@<sql_directory>/sql/create_log_indexes.sql
```

11. Create the Tivoli Workload Scheduler indexes, by running the following command:

```sql
sqlplus -L -S <TWS_MDL_USER>/TWS_MDL_PW
@<ORACLE_NET_SERVICE_NAME>
@<sql_directory>/sql/create_indexes.sql
```

12. Create the Tivoli Workload Scheduler event rule views, by running the following command:

```sql
sqlplus -L -S <TWS_MDL_USER>/TWS_MDL_PW
@<ORACLE_NET_SERVICE_NAME>
@<sql_directory>/sql/create_rule_views.sql
```

13. Create the Tivoli Workload Scheduler log views, by running the following command:

```sql
sqlplus -L -S <TWS_MDL_USER>/TWS_MDL_PW
@<ORACLE_NET_SERVICE_NAME>
@<sql_directory>/sql/create_log_views.sql
```

14. Create the Tivoli Workload Scheduler views, by running the following command:

```sql
sqlplus -L -S <TWS_MDL_USER>/TWS_MDL_PW
@<ORACLE_NET_SERVICE_NAME>
@<sql_directory>/sql/create_views.sql
```

15. Create the job repository, by running the following command:

```sql
sqlplus -L -S <TWS_MDL_USER>/TWS_MDL_PW
@<ORACLE_NET_SERVICE_NAME>
@<sql_directory>/DWB/sql/create_job_repository.sql
```

16. Create the allocation repository, by running the following command:

```sql
sqlplus -L -S <TWS_MDL_USER>/TWS_MDL_PW
@<ORACLE_NET_SERVICE_NAME>
@<sql_directory>/DWB/sql/create_allocation_repository.sql
```

17. Create the resource repository, by running the following command:

```sql
sqlplus -L -S <TWS_MDL_USER>/TWS_MDL_PW
@<ORACLE_NET_SERVICE_NAME>
@<sql_directory>/DWB/sql/create_resource_repository.sql
```

18. Create the server table, by running the following command:

```sql
sqlplus -L -S <TWS_MDL_USER>/TWS_MDL_PW
@<ORACLE_NET_SERVICE_NAME>
@<sql_directory>/DWB/sql/create_server.sql
```
Before installing a dynamic domain manager: If you are creating the SQL tables before installing a dynamic domain manager, perform the following steps:

1. Create the Tivoli Workload Scheduler users, by running the following command:

   sqlplus -L -S `<ORACLE_ADMIN>/`<ORACLE_ADMIN_PW>`
   @`<ORACLE_NET_SERVICE_NAME>`
   @`<sql_directory>/DWB/sql/create_users.sql`

   where:

   **ORACLE_NET_SERVICE_NAME**

   The name used by clients to identify an Oracle Net server and the specific system identifier or database for the Oracle Net connection. A net service name is mapped to a port number and protocol. It is also known as a connect string, database alias, host string, or service name.

   When you install the IT administrator must specify this value in the Net service name field.

   If your Oracle database will be:
   - Installed on the same system where you are installing your master domain manager or a backup master, the net service name is the name of your Oracle database.
   - Installed on the same system where you are installing your dynamic domain manager or a backup dynamic domain manager, the net service name is the name of your Oracle database.
   - Not installed on the system where you are installing your master domain manager or a backup master, the net service name is the alias configured for the connection to the remote database.
   - Not installed on the system where you are installing your dynamic domain manager or a backup dynamic domain manager, the net service name is the alias configured for the connection to the remote database.

   **sql_directory**

   The directory where you stored the SQL you created in "Generating the SQL files for DB2" on page 39.

2. Create the job repository, by running the following command:

   sqlplus -L -S `<TWS_MDL_USER>/`<TWS_MDL_PW>`
   @`<ORACLE_NET_SERVICE_NAME>`
   @`<sql_directory>/DWB/sql/create_job_repository.sql`

3. Create the allocation repository, by running the following command:

   sqlplus -L -S `<TWS_MDL_USER>/`<TWS_MDL_PW>`
   @`<ORACLE_NET_SERVICE_NAME>`
   @`<sql_directory>/DWB/sql/create_allocation_repository.sql`

4. Create the resource repository, by running the following command:

   sqlplus -L -S `<TWS_MDL_USER>/`<TWS_MDL_PW>`
   @`<ORACLE_NET_SERVICE_NAME>`
   @`<sql_directory>/DWB/sql/create_resource_repository.sql`

5. Create the server table, by running the following command:

   sqlplus -L -S `<TWS_MDL_USER>/`<TWS_MDL_PW>`
   @`<ORACLE_NET_SERVICE_NAME>`
   @`<sql_directory>/DWB/sql/create_server.sql`
Before upgrading Tivoli Workload Scheduler from V8.3 and later to V8.5: If you are upgrading the SQL tables before upgrading Tivoli Workload Scheduler from V8.3 and later to V8.5, perform the following steps:

1. Log on to the ORACLE database with the user you specified in the MDL_USER property.

2. Verify the version of your database, by running the following command:
   
   ```
   select * FROM <MDL_USER>.MPR_MODEL_PROPERTIES
   where MPR_NAME='DATABASE_VERSION'
   ```

   If `DATABASE_VERSION` returns one of the following values:
   
   - **8.3.0.00**
     - You have installed version 8.3.
   - **8.3.0.01**
     - You have installed version 8.3.0.1.
   - **8.4.0.00**
     - You have installed version 8.4.
   - **8.4.0.01**
     - You have installed version 8.4.0.1.
   - **8.5.0.00**
     - You have installed version 8.5.

   run the commands listed in Step [3](#).

   If `DATABASE_VERSION` returns one of the following values:
   
   - **8.5.1.00**
     - You have installed version 8.5.1.
   - **8.5.1.01**
     - You have installed version 8.5.1.1.

   run the procedure described in ["Before upgrading from V8.5.1 to V8.5.1.1" on page 57](#).

3. Run the following commands in sequence starting with the command for your specific database version considering if you used or not the ORACLE partitioning feature.

   For example, if `DATABASE_VERSION` is equal to 8.5.0.00 and you:

   **Did not use the ORACLE partitioning feature:**
   Run the following commands:
   
   a. 4d on page 54
   b. 4e on page 54
   c. 4f on page 54
   d. 4g on page 54

   **Used the ORACLE partitioning feature:**
   Run the following commands:
   
   a. 7d on page 55
   b. 7e on page 55
   c. 7f on page 55
   d. 7g on page 55

   **To upgrade the Tivoli Workload Scheduler database tables, if you did not use the ORACLE partitioning feature:**
   Run the following commands:
a. sqlplus -L -S <TWS_MDL_USER>/TWS_MDL_PW>
   @<ORACLE_NET_SERVICE_NAME>
   @<sql_directory>/sql/stded/upgrade_83001_84000.sql

**ORACLE_NET_SERVICE_NAME**

The name used by clients to identify an Oracle Net server and the specific system identifier or database for the Oracle Net connection. A net service name is mapped to a port number and protocol. It is also known as a connect string, database alias, host string, or service name. When you install the IT administrator must specify this value in the **Net service name** field.

If your Oracle database will be:

- Installed on the same system where you are installing your master domain manager or a backup master, the net service name is the name of your Oracle database.
- Installed on the same system where you are installing your dynamic domain manager or a backup dynamic domain manager, the net service name is the name of your Oracle database.
- Not installed on the system where you are installing your master domain manager or a backup master, the net service name is the alias configured for the connection to the remote database.
- Not installed on the system where you are installing your dynamic domain manager or a backup dynamic domain manager, the net service name is the alias configured for the connection to the remote database.

**sql_directory**

The directory where you stored the SQL you created in "Generating the SQL files for DB2" on page 39.

b. sqlplus -L -S <TWS_MDL_USER>/TWS_MDL_PW>
   @<ORACLE_NET_SERVICE_NAME>
   @<sql_directory>/sql/upgrade_84001_84001.sql

c. sqlplus -L -S <TWS_MDL_USER>/TWS_MDL_PW>
   @<ORACLE_NET_SERVICE_NAME>
   @<sql_directory>/sql/stded/upgrade_84001_85000.sql

d. sqlplus -L -S <TWS_MDL_USER>/TWS_MDL_PW>
   @<ORACLE_NET_SERVICE_NAME>
   @<sql_directory>/sql/upgrade_85000_85100.sql

e. sqlplus -L -S <TWS_MDL_USER>/TWS_MDL_PW>
   @<ORACLE_NET_SERVICE_NAME>
   @<sql_directory>/sql/upgrade_85100_85101.sql

f. sqlplus -L -S <TWS_MDL_USER>/TWS_MDL_PW>
   @<ORACLE_NET_SERVICE_NAME>
   @<sql_directory>/sql/stded/upgrade_85101_86000.sql

g. sqlplus -L -S <TWS_MDL_USER>/TWS_MDL_PW>
   @<ORACLE_NET_SERVICE_NAME>
   @<sql_directory>/sql/check_forecast.sql

- If the command returns a value different from 0 run the following command:
  sqlplus -L -S <TWS_MDL_USER>/TWS_MDL_PW>
  @<ORACLE_NET_SERVICE_NAME>
  @<sql_directory>/sql/upgrade_84001_84005.sql
To upgrade the Tivoli Workload Scheduler database tables, if you used the ORACLE partitioning feature:

Run the following commands:

a. `sqlplus -L -S <TWS_MDL_USER>/<TWS_MDL_PW>
   @ORACLE_NET_SERVICE_NAME
   @<sql_directory>/sql/upgrade_83001_84000.sql`

*ORACLE_NET_SERVICE_NAME*

The name used by clients to identify an Oracle Net server and the specific system identifier or database for the Oracle Net connection. A net service name is mapped to a port number and protocol. It is also known as a connect string, database alias, host string, or service name.

When you install the IT administrator must specify this value in the **Net service name** field.

If your Oracle database will be:

- Installed on the same system where you are installing your master domain manager or a backup master, the net service name is the name of your Oracle database.
- Installed on the same system where you are installing your dynamic domain manager or a backup dynamic domain manager, the net service name is the name of your Oracle database.
- Not installed on the system where you are installing your master domain manager or a backup master, the net service name is the alias configured for the connection to the remote database.
- Not installed on the system where you are installing your dynamic domain manager or a backup dynamic domain manager, the net service name is the alias configured for the connection to the remote database.

*sql_directory*

The directory where you stored the SQL you created in "Generating the SQL files for DB2" on page 39.

b. `sqlplus -L -S <TWS_MDL_USER>/<TWS_MDL_PW>
   @ORACLE_NET_SERVICE_NAME
   @<sql_directory>/sql/upgrade_84001_84001.sql`
c. `sqlplus -L -S <TWS_MDL_USER>/<TWS_MDL_PW>
   @ORACLE_NET_SERVICE_NAME
   @<sql_directory>/sql/upgrade_84001_85000.sql`
d. `sqlplus -L -S <TWS_MDL_USER>/<TWS_MDL_PW>
   @ORACLE_NET_SERVICE_NAME
   @<sql_directory>/sql/upgrade_85000_85100.sql`
e. `sqlplus -L -S <TWS_MDL_USER>/<TWS_MDL_PW>
   @ORACLE_NET_SERVICE_NAME
   @<sql_directory>/sql/upgrade_85100_85101.sql`
f. `sqlplus -L -S <TWS_MDL_USER>/<TWS_MDL_PW>
   @ORACLE_NET_SERVICE_NAME
   @<sql_directory>/sql/upgrade_85101_86000.sql`
g. `sqlplus -L -S <TWS_MDL_USER>/<TWS_MDL_PW>
   @ORACLE_NET_SERVICE_NAME
   @<sql_directory>/sql/check_forecast.sql`

- If the command returns a value different from 0 run the following command:
To upgrade the dynamic workload broker database tables:

a. Create the job repository, by running the following command:
   ```
   sqlplus -L -S <TWS_MDL_USER>/<TWS_MDL_PW>
   @<ORACLE_NET_SERVICE_NAME>
   @<sql_directory>/DWB/sql/create_job_repository.sql
   ```

The name used by clients to identify an Oracle Net server and the specific system identifier or database for the Oracle Net connection. A net service name is mapped to a port number and protocol. It is also known as a connect string, database alias, host string, or service name.

When you install the IT administrator must specify this value in the Net service name field.

If your Oracle database will be:

- Installed on the same system where you are installing your master domain manager or a backup master, the net service name is the name of your Oracle database.
- Installed on the same system where you are installing your dynamic domain manager or a backup dynamic domain manager, the net service name is the name of your Oracle database.
- Not installed on the system where you are installing your master domain manager or a backup master, the net service name is the alias configured for the connection to the remote database.
- Not installed on the system where you are installing your dynamic domain manager or a backup dynamic domain manager, the net service name is the alias configured for the connection to the remote database.

b. Create the allocation repository, by running the following command:
   ```
   sqlplus -L -S <TWS_MDL_USER>/<TWS_MDL_PW>
   @<ORACLE_NET_SERVICE_NAME>
   @<sql_directory>/DWB/sql/create_allocation_repository.sql
   ```

c. Create the resource repository, by running the following command:
   ```
   sqlplus -L -S <TWS_MDL_USER>/<TWS_MDL_PW>
   @<ORACLE_NET_SERVICE_NAME>
   @<sql_directory>/DWB/sql/create_resource_repository.sql
   ```

d. Create the server table, by running the following command:
   ```
   sqlplus -L -S <TWS_MDL_USER>/<TWS_MDL_PW>
   @<ORACLE_NET_SERVICE_NAME>
   @<sql_directory>/DWB/sql/create_server.sql
   ```

e. Log on to the ORACLE database as SYSTEM and run the following command:
   ```
   grant create procedure to <MDL_USER>
   ```
**Before upgrading from V8.5.1 to V8.5.1.1:** If you are upgrading the SQL tables before upgrading Tivoli Workload Scheduler from version 8.5.1 to version 8.5.1.1, perform the following steps:

1. Log on to the ORACLE database with the user you specified in the `MDL_USER` property.
2. Verify the version of your database, by running the following command:
   ```sql
   select * FROM <MDL_USER>.MDL_MODEL_PROPERTIES
   where MPR_NAME='DATABASE_VERSION'
   ```
   If `DATABASE_VERSION` returns one of the following values:
   - **8.3.0.00**
     You have installed version 8.3.
   - **8.3.0.01**
     You have installed version 8.3.0.1.
   - **8.4.0.00**
     You have installed version 8.4.
   - **8.4.0.01**
     You have installed version 8.4.0.1.
   - **8.5.0.00**
     You have installed version 8.5.
   
   run the procedure described in "Before upgrading Tivoli Workload Scheduler from V8.3 and later to V8.5" on page 53.
   
   If `DATABASE_VERSION` returns one of the following values:
   - **8.5.1.00**
     You have installed version 8.5.1.
   - **8.5.1.01**
     You have installed version 8.5.1.1.
   
   run the procedure described in Step 3.

3. Run the following commands in sequence starting with the command for your specific database version, depending on whether or not you used the ORACLE partitioning feature:

   **To upgrade the Tivoli Workload Scheduler database tables, if you did not use the ORACLE partitioning feature:**
   ```sql
   sqlplus -L -S <TWS_MDL_USER>/<TWS_MDL_PW>
   @<ORACLE_NET_SERVICE_NAME>/sql/upgrade_85100_85101.sql
   ```

   `ORACLE_NET_SERVICE_NAME`
   The name used by clients to identify an Oracle Net server and the specific system identifier or database for the Oracle Net connection. A net service name is mapped to a port number and protocol. It is also known as a connect string, database alias, host string, or service name.

   When you install the IT administrator must specify this value in the Net service name field.

   If your Oracle database will be:

   - Installed on the same system where you are installing your master domain manager or a backup master, the net service name is the name of your Oracle database.
- Installed on the same system where you are installing your dynamic domain manager or a backup dynamic domain manager, the net service name is the name of your Oracle database.

- Not installed on the system where you are installing your master domain manager or a backup master, the net service name is the alias configured for the connection to the remote database.

- Not installed on the system where you are installing your dynamic domain manager or a backup dynamic domain manager, the net service name is the alias configured for the connection to the remote database.

```
sql_directory
The directory where you stored the SQL you created in
"Generating the SQL files for DB2" on page 39.
```

b. `sqlplus -L -S `<TWS_MDL_USER>`/`<TWS_MDL_PW>`
@`<ORACLE_NET_SERVICE_NAME>`
@`<sql_directory>`/sql/stded/upgrade_85101_86000.sql

To upgrade the Tivoli Workload Scheduler database tables, if you used the ORACLE partitioning feature:

a. `sqlplus -L -S `<TWS_MDL_USER>`/`<TWS_MDL_PW>`
@`<ORACLE_NET_SERVICE_NAME>`
@`<sql_directory>`/sql/upgrade_85100_85101.sql

```
ORACLE_NET_SERVICE_NAME
The name used by clients to identify an Oracle Net server and the specific system identifier or database for the Oracle Net connection. A net service name is mapped to a port number and protocol. It is also known as a connect string, database alias, host string, or service name.
```

When you install the IT administrator must specify this value in the Net service name field.

If your Oracle database will be:

- Installed on the same system where you are installing your master domain manager or a backup master, the net service name is the name of your Oracle database.

- Installed on the same system where you are installing your dynamic domain manager or a backup dynamic domain manager, the net service name is the name of your Oracle database.

- Not installed on the system where you are installing your master domain manager or a backup master, the net service name is the alias configured for the connection to the remote database.

- Not installed on the system where you are installing your dynamic domain manager or a backup dynamic domain manager, the net service name is the alias configured for the connection to the remote database.

```
sql_directory
The directory where you stored the SQL you created in
"Generating the SQL files for DB2" on page 39.
```
To upgrade the dynamic workload broker database tables:

```sql
sqlplus -L -S <TWS_MDL_USER>/<TWS_MDL_PW>
@<ORACLE_NET_SERVICE_NAME>
@<sql_directory>/DWB/sql/upgrade_85101_86000.sql
```

### Installing

This chapter describes how to perform a first-time installation of the current version of Tivoli Workload Scheduler. It contains the following sections:

- “User authorization requirements”
- “Installing DB2”
- “Using the installation wizard” on page 60
- “Performing a silent installation” on page 84
- “Installing agents using twsinst” on page 88
- “Installing agents using Software Distribution” on page 96
- “Installing the Job Brokering Definition Console” on page 105
- “Installing the additional plug-ins by using the Tivoli Workload Scheduler for Additional Plug-ins” on page 106

### User authorization requirements

Check the authorization roles before beginning the procedure.

Authorization requirements for running any of the installation, upgrade, or uninstallation wizards or commands are the same:

**UNIX and Linux operating systems**

root access

**Windows operating system**

Your login account must be a member of the Windows **Administrators** group or domain administrators with **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**. To do this, before running the installation, perform the following steps:

1. Right-click the icon that you use to run the program:
   - **If you are using the wizard:**
     SETUP.exe
   - **If you are using the silent installation or twsinst:**
     Command Prompt

2. Select **Run as administrator**.

In addition, to use Software Distribution, the user must have the Software Distribution roles **admin**, **senior**, or **super**.

### Installing DB2

For detailed information about DB2, see the DB2 documentation at [http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp](http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp)

To install DB2, choose one of the following options:
Using the installation wizard

This section describes how to install Tivoli Workload Scheduler components using the installation wizard. The installation wizard runs on all supported operating systems. For a complete list of supported operating systems, see "http://www.ibm.com/support/docview.wss?rs=672&uid=swg27012175"

Note: IBM i is an exception. The dynamic agent on this platform can be installed only using the \( \text{twsinst} \) command line.

Installing a master domain manager or backup master

For a graphical installation, from the installation DVD or from the appropriate image, start the launchpad as described in "Launchpad" on page 30 and select the Tivoli Workload Scheduler installation, or run the setup for the operating system on which you are installing:

On Windows operating systems

\( \text{TWS}\backslash\text{operating\_system}\backslash\text{SETUP}.exe \) or \( \text{TWS}\backslash\text{SETUP}.cmd \)

On UNIX and Linux operating systems

\( \text{TWS}\backslash\text{operating\_system}\backslash\text{SETUP}.bin \) or \( \text{TWS}\backslash\text{SETUP}.sh \)

Note: Your RDBMS must be running when you begin the installation.

If you are installing on UNIX or Linux and you want the installation to stop if the Tivoli Workload Scheduler prerequisite check encounters an error or warning, specify the parameter \( -W\ \text{checkPrerequisites}.stopOnCheckPrereq=true \) after the setup command. See "Checking prerequisites (UNIX and Linux)" on page 24 for more information about the prerequisite check.

When you install a master domain manager and a backup master domain manager the following workstation types are created in the database:

- **master**
  - For the master domain manager

- **broker**
  - For the broker server

- **agent**
  - For the dynamic agent

If you are installing a backup master domain manager, you must configure it manually after you perform the installation. See "Configuring a backup master domain manager" on page 161.
The installation steps for the master domain manager or the backup master
domain manager installation process are described in the following sections:

1. “Tivoli Workload Scheduler installation options”
2. “WebSphere Application Server installation options” on page 64
3. “RDBMS installation options” on page 65

This section is divided into the following subsections:

• “Installing for a DB2 database server” on page 65
• “Installing for a DB2 database client” on page 67
• “Installing for an Oracle database” on page 69

4. “Dynamic workload broker configuration information” on page 71

After a successful installation, perform one of the following configuration tasks,
depending on the type of component you installed:

• “Configuring a master domain manager” on page 160
• “Configuring a backup master domain manager” on page 161.

### Tivoli Workload Scheduler installation options:
Complete the following Tivoli Workload Scheduler data fields.

#### User name

Specify the Tivoli Workload Scheduler user name. 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.

**On Windows operating systems**

- If this user account does not already exist, it is automatically
  created by the installation wizard.
- 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. Create a user with a home directory and group. By
default, Tivoli Workload Scheduler is installed in TWA_home

**Note:** If you are installing into a new instance of Tivoli Workload
Automation, the Tivoli Workload Scheduler user name and
password are also used as the WebSphere Application Server
administrator user name and password.

#### Password

Specify the Tivoli Workload Scheduler password. The password must
comply with the password policy in your Local Security Settings. Spaces
are not permitted.

**On Windows operating systems**

Password for users can include alphanumeric, dash (-), underscore
(_) characters, and ()!?=^*/~\[] $`+;.: ,@.

**On UNIX and LINUX systems**

Password for users can include any alphanumeric, dash (-),
underscore (_) characters, and ()!?=^*/~+. 

Master domain manager and backup master domain manager configuration information

Company
The name of the company. Spaces are allowed and the maximum field length is 40 characters.

This workstation name
The name of the workstation where you are installing the instance. The default is the host name of the workstation.
When installing a master domain manager, the name you specify here is the name of the Tivoli Workload Scheduler workstation known in the database as master.
When installing a backup master domain manager, the name you specify here is the name of the Tivoli Workload Scheduler workstation known in the database as fta. Spaces are not allowed and the maximum field length is 16 characters. If the host name is longer than 16 characters, an alternative name must be provided for a successful installation. It can contain alphanumeric, dash (-), and underscore (_) characters. The first character must be a letter.

Master domain manager name
The name of the master domain manager workstation. This field is required if you are installing a backup manager. If you are not installing a backup manager, this field is grayed out. Spaces are not allowed and the maximum field length is 16 characters. The first character cannot be numeric.

Tivoli Workload Scheduler Netman port
The port used by the Netman process to run distributed scheduling. Netman is the network process that controls the production environment. The default value is 31111. The valid range is from 1 to 65535.

Note: If you change this value, all default port number values in the application server port information panel are changed to reflect the new range. For example, if you specify 42111 as TCP/IP port number, the default for HTTP transport becomes 42125, the default for HTTPS becomes 42126, and so on.

Dynamic agent configuration information

Host name or IP address
The fully qualified host name or IP address of the dynamic agent. The dynamic workload broker and the Tivoli Workload Scheduler for z/OS controller use this address to connect to the dynamic agent.

Agent display name
The name of the dynamic agent workstation definition.

JobManager port number
The dynamic agent secure port number (SECUREADDR). The Tivoli Workload Scheduler for z/OS controller and the dynamic workload broker use this port to contact the Tivoli Workload Scheduler dynamic agent. The default value is 31114. The valid range is from 1 to 65535.
Dynamic workload broker host name
Applies only to backup master domain manager. The fully qualified hostname the dynamic workload broker. The Tivoli Workload Scheduler for z/OS controller and the dynamic agent use this address to contact the dynamic workload broker.

Dynamic workload broker HTTPS port number
Applies only to backup master domain manager. The HTTPS transport port of the dynamic workload broker. The dynamic agent uses this port to connect to the dynamic workload broker. The default value is 31116 although if you leave the field blank, it defaults to 0. The valid range is from 1 to 65535.

Add the "FINAL" job stream to the database to automate the production cycle
This option is available only if you are installing a master domain manager. To add the final job stream to the database. This option allows automatic production plan extension at the end of each current production plan processing. By default, this box remains unchecked.

Note: During the installation, if you identified an existing Tivoli Workload Scheduler database that has a final job stream, the installation does not overwrite it.

Automatically generate all the other Tivoli Workload Scheduler ports
By default, this box is checked. If you leave this box selected, all the ports needed by WebSphere Application Server are automatically generated using the default values and the application server port information panel is not displayed. The installation procedure checks for the availability of the ports in the specified port range. If one or more ports are in use by other applications, you are prompted to enter a new port number. If you have not requested to generate ports automatically, specify the values for the ports used by the application server embedded in the Tivoli Workload Scheduler instance. Accept the default values unless you know that they are already in use by other applications.

Installation directory
Enter the name of the directory where the Tivoli Workload Scheduler instance is installed for the specified user. The maximum field length is 46 characters and the name must not contain numbers. Parentheses () are not allowed. You cannot use national characters.

Spaces are allowed. However, you cannot install Tivoli Workload Scheduler for Applications version 8.2.1 or earlier on the current version of Tivoli Workload Scheduler if the directory path contains spaces.

On Windows operating systems:
- The name must be longer than three characters, the second character must be :, and the third character must be \
- The default directory is %ProgramFiles%\IBM\TWA.

On UNIX and Linux systems:
- The name must be longer than one character and the first character must be /
- The default directory is the /opt/IBM/TWA directory.
- Optionally check Create symbolic links to create links in the /usr/bin directory. Any existing Tivoli Workload Scheduler symbolic links are overwritten.
**WebSphere Application Server installation options:**
The following fields are provided for WebSphere Application Server data. The fields you complete depend upon whether you are installing into a new instance of Tivoli Workload Automation or into an existing instance. The installation procedure checks for the availability of the ports in the specified port range. If one or more ports are in use by other applications, you are prompted to enter a new port number.

**New instance**
If you are installing into a new instance of Tivoli Workload Automation, provide the following information.

**HTTP transport**
The port for the HTTP transport. It is used by the composer command line interface when this protocol is selected. The default value is **31115**. The valid range is from 1 to 65535.

**HTTPS transport**
The port for the secure HTTP transport. It is used by the composer command line interface when this protocol is selected. The default value is **31116**. The valid range is from 1 to 65535.

**Bootstrap**
The port for the bootstrap or RMI. It is used by the graphical user interfaces. The default value is **31117**. The valid range is from 1 to 65535.

**SOAP connector**
The port for the application server protocol SOAP connector. The default value is **31118**. The valid range is from 1 to 65535.

**SAS Server Authentication Listener**
The port used by the Secure Association Services (SAS) to listen for inbound authentication requests. The default value is **31119**. The valid range is from 1 to 65535.

**CSIv2 Server Authentication Listener**
The port on which the Common Secure Interoperability Version 2 (CSIv2) service listens for inbound server authentication requests. The default value is **31120**. The valid range is from 1 to 65535.

**CSIv2 Client Authentication Listener**
The port on which the Common Secure Interoperability Version 2 (CSIv2) service listens for inbound client authentication requests. The default value is **31121**. The valid range is from 1 to 65535.

**ORB Listener**
The port used for RMI over IIOP communication. The default value is **31122**. The valid range is from 1 to 65535.

**Administration HTTP transport**
The administrative console port. The default value is **31123**. The valid range is from 1 to 65535.

**Administration HTTPS transport**
The administrative console secure port. The default value is **31124**. The valid range is from 1 to 65535.

**Event Processor**
The port used by the event management processor to receive
events. The default value is 31131. The valid range is from 1 to 65535. This parameter is not requested if you are installing a backup master domain manager.

Existing instance
If you are installing into an existing instance of Tivoli Workload Automation, you are only required to provide the WebSphere Application Server administrator user name and password. The port data is automatically retrieved from the existing WebSphere Application Server instance. If you do not know the user name, you can click Retrieve on the appropriate panel to populate this field, although you must still provide the password. If you click Retrieve, you might have to wait for the field to populate.

RDBMS installation options:
This section is divided into subsections. See the section that corresponds to the RDBMS you are using.

- "Installing for a DB2 database server"
- "Installing for a DB2 database client" on page 67
- "Installing for an Oracle database" on page 69

Installing for a DB2 database server:
When you are installing into an existing database, perform the steps described in "Tivoli Workload Scheduler installation options" on page 61. The following list describes the fields that you might need to complete during the installation.

DB2 search path
Type or Browse for the directory where the existing DB2 instance is installed. On Windows, the default is %ProgramFiles%\IBM\sql\lib. If you have more than one DB2 instance installed, make sure you provide the fully qualified path to the DB2 instance you want. This path must identify a tree in the DB2 structure that includes the db2level.exe file.

Instance name
The name of the DB2 server instance.

Instance port
The TCP/IP port number used to communicate with the DB2 instance. The default is 50000.

DB2 server administrator user
The user name of the administrator of the DB2 server instance. This user can also be any user having SYSADM or SYSCTRL authority on the DB2 server. On UNIX, verify that you can switch to this user and that it can load the DB2 environment.

If the DB2 administrator already created the database tables using the "Creating or upgrading the database tables if you are using DB2" on page 37 procedure, the user name is the one that the DB2 administrator specified in the DB_USER property in the customize0DB$SQL.properties file.

On Windows operating systems
The default value is db2admin.

On UNIX and Linux operating systems
The default value is db2inst1.
DB2 server administrator password
The password of the DB2 server administrator user, or of the user with SYSADM or SYSCTRL authority. You are asked to confirm the password.

Database name
The name of the DB2 database. The maximum length is five characters. You can use an existing DB2 database instance if its name does not exceed five characters. When you are installing:

Master domain manager
Provide the name of a database that is not used by a dynamic domain manager.

Backup master domain manager
Provide the name of the master domain manager database.

Dynamic domain manager
Provide the name of a database that is not used by a master domain manager.

Backup dynamic domain manager
Provide the name of the dynamic domain manager database.

For information about DB2 database names, see the DB2 documentation.

Specify advanced configuration parameters for the Tivoli Workload Scheduler database
Select this option if you want to specify the following advanced parameters:

Tablespace name
The name of the DB2 instance tablespace. This tablespace is used to store scheduling objects and event rules. For information about DB2 table spaces, see the DB2 documentation.

Tablespace path
The relative path of the DB2 table space. The path can be a relative or a fully qualified path. When the table space path is a fully qualified path, the DB2 administrator user must have complete access rights to the directory where the table space is installed. For more information see “DB2 tablespace relative paths” on page 312.

The default table space path name is TWS_DATA. The default table space temporary directory is TWS_TEMP. For UNIX and Linux operating systems, make sure that the DB2 Administrator has write access to the directory above the table space directory.

Tablespace used to store event logs
Specify the name and path of the DB2 table space where Tivoli Workload Scheduler event logs are to be stored. These logs include data about event rule instances, triggered actions, and operator messages displayed by the Dynamic Workload Console. Data from the logs can be used to create reports. You can view report data using the Dynamic Workload Console.

Report tablespace name
The name of the table space for storing report data. The default name is TWS_LOG.

Report tablespace path
The path of the table space for storing report data. The default path is TWS_LOG. The path can be a relative or a fully qualified
When the table space path is a fully qualified path the DB2 administrator user must have complete access rights to the directory where the table space is installed. For more information, see "DB2 tablespace relative paths" on page 312. Note that the report tablespace path cannot be the same as the tablespace path.

Installing for a DB2 database client:
During the installation of the backup master domain manager, you install a DB2 client to connect to the DB2 server that contains the Tivoli Workload Scheduler database. This database was created by the master domain manager installation. If it is a DB2 database server, the database is on the workstation of the master domain manager. If it is a DB2 database client, the database is on another workstation.

During the installation of the backup dynamic domain manager, you install a DB2 client to connect to the DB2 server that contains the Tivoli Workload Scheduler database. This database was created by the dynamic domain manager installation. If it is a DB2 database server, the database is on the workstation of the dynamic domain manager. If it is a DB2 database client, the database is on another workstation.

When you are installing with an existing database, perform the steps described in "Tivoli Workload Scheduler installation options" on page 61. The following list describes the fields that you might need to complete during the installation.

DB2 search path
Type or Browse for the directory where the existing DB2 instance is installed. If you have more than one DB2 instance installed, make sure you provide the fully qualified path to the DB2 instance you want. This path must identify a tree in the DB2 structure that includes the db2level.exe file.

Remote database server
The IP address or host name of the workstation where the DB2 server is installed.

Remote database port
The TCP/IP port number that the remote DB2 server instance uses to communicate.

Identify the user on the remote DB2 server to be used by the installation for DB2 administration tasks
Provide the following data:

DB2 server administrator user
The user name of the administrator of the DB2 server instance. This user can also be any user having SYSADM or SYSCTRL authority on the DB2 server. On UNIX, verify that you can switch to this user and that it can load the DB2 environment.

If the DB2 administrator already created the database tables using the "Creating or upgrading the database tables if you are using DB2" on page 37 procedure, the user name is the one that the DB2 administrator specified in the DB_USER property in the customizeDB2SQL.properties file.

On Windows operating systems
The default value is db2admin.
On UNIX and Linux operating systems
The default value is db2inst1.

DB2 server administrator password
The password of the DB2 server administrator user, or of the user with SYSADM or SYSCTRL authority. You are asked to confirm the password.

Identify the user on the DB2 client to be used by the installation for DB2 administration tasks
Specify the user on the DB2 client to be used by the installation for DB2 administration tasks. Provide the following data:

DB2 client administrator user
The user name of the DB2 administrator of the DB2 client instance. The user ID must contain the following login properties:
- \( \text{login} = \text{true} \)
- \( \text{rlogin} = \text{true} \)

DB2 client administrator password
The password of the DB2 administrator of the DB2 client instance.

Note: The password must comply with the password policy in your Local Security Settings, otherwise the installation fails.

Identify the user on the DB2 server to be used by Tivoli Workload Scheduler to access the database, if different from the DB2 Server Administration User
Select this option when the DB2 server user used to access Tivoli Workload Scheduler is different from the DB2 Server Administration User. Provide the following data:

Tivoli Workload Scheduler DB2 user
The user name of the Tivoli Workload Scheduler DB2 user.

Tivoli Workload Scheduler DB2 password
The password of the Tivoli Workload Scheduler DB2 user.

Database name
The name of the DB2 database. The maximum length is five characters. You can use an existing DB2 database instance if its name does not exceed five characters. When you are installing a:

Master domain manager
Provide the name of a database that is not used by a dynamic domain manager.

Backup master
Provide the name of the master domain manager database.

Dynamic domain manager
Provide the name of a database that is not used by a master domain manager.

Backup dynamic domain manager
Provide the name of the dynamic domain manager database.
For information about DB2 database names, see the DB2 documentation.

Specify advanced configuration parameters for the Tivoli Workload Scheduler database
Select this option if you want to specify the following advanced parameters:

Tablespace name
The name of the DB2 instance table space. For information about DB2 table spaces, see the DB2 documentation.

Tablespace path
The relative path of the DB2 table space. The path can be a relative or a fully qualified path. When the table space path is a fully qualified path, the DB2 administrator user must have complete access rights to the directory where the table space is installed. For more information, see “DB2 tablespace relative paths” on page 312.

The default table space path name is TWS_DATA. The default table space temporary directory is TWS_TEMP. For UNIX and Linux operating systems, make sure that the DB2 Administrator has write access to the directory above the table space directory.

Tablespace used to store event logs
Specify the name and path of the DB2 table space where Tivoli Workload Scheduler event logs are to be stored. These logs are used to create reports. You can view report data using the Dynamic Workload Console.

Report tablespace name
The name of the table space for storing report data. The default name is TWS_LOG.

Report tablespace path
The path of the table space for storing report data. The default path is TWS_LOG. The path can be a relative or a fully qualified path. When the table space path is a fully qualified path, the DB2 administrator user must have complete access rights to the directory where the table space is installed. For more information, see “DB2 tablespace relative paths” on page 312.

Installing for an Oracle database:

When you are installing for an Oracle database both for server and client, follow the installation wizard prompts. The following list describes the fields that you might need to complete during the installation.

Oracle Database search path
Specify the path of an Oracle installation that satisfies the Tivoli Workload Scheduler prerequisites. The fully qualified path must identify a tree in the Oracle structure that includes the sqlplus executable.

Net service name
The name used by clients to identify an Oracle Net server and the specific system identifier or database for the Oracle Net connection. A net service name is mapped to a port number and protocol. It is also known as a connect string, database alias, host string, or service name.
If your Oracle database is:
- Installed on the same system where you are installing your master domain manager or a backup master, the net service name is the name of your Oracle database.
- Installed on the same system where you are installing your dynamic domain manager or a backup dynamic domain manager, the net service name is the name of your Oracle database.
- Not installed on the system where you are installing your master domain manager or a backup master, the net service name is the alias configured for the connection to the remote database.
- Not installed on the system where you are installing your dynamic domain manager or a backup dynamic domain manager, the net service name is the alias configured for the connection to the remote database.

Contact your database administrator to obtain the correct net service name.

**Oracle administrator user**
The database administrator user name (such as `SYSTEM`) required to authenticate to the Oracle database. This account must already exist.

If the ORACLE administrator already created the database tables using the “Creating or upgrading the database tables if you are using Oracle” on page 47, the user name is the one that the ORACLE administrator specified in the `MDL_USER` property of the `customizeORACLESQL.properties` file.

**Oracle administrator user password**
The database administrator user password required to authenticate to the Oracle database.

**Tivoli Workload Scheduler Oracle user**
The owner of the Tivoli Workload Scheduler schema.

If the ORACLE administrator already created the database tables using the “Creating or upgrading the database tables if you are using Oracle” on page 47, the user name is the one that the ORACLE administrator specified in the `MDL_USER` property of the `customizeORACLESQL.properties` file. The name must comply with the Oracle naming rules.

If you are installing a:

**Master domain manager**
If you leave this field blank, this name is defaulted to `<TWS_user>`.

**Backup master**
Enter the same name that you used in the master domain manager.

**Dynamic domain manager**
If you leave this field blank, this name is defaulted to `<TWS_user>`.
Provide a name different from the one that you used when installing the master domain manager.

**Backup dynamic domain manager**
Enter the same name that you used in the dynamic domain manager.

On a fresh installation of a:

**Master domain manager**
This user does not exist in the database. If this is not the case, it means that there already is a master domain manager or a backup master instance pointing to the same database with this user name.
If your existing Tivoli Workload Scheduler instance is version 8.3 or later, the installation process upgrades the current database schema to the new schema.

**Dynamic domain manager**

This user does not exist in the database. If this is not the case, it means that there already is a dynamic domain manager or a backup dynamic domain manager pointing to the same database with this user name. If your existing Tivoli Workload Scheduler instance is version 8.3 or later, the installation process upgrades the current database schema to the new schema.

If your existing instance is the current version, the installation process assumes that the schema is at the correct level and does not create the database objects (tables, views, clusters, procedures, indexes, and so on) for Tivoli Workload Scheduler.

If you identify an existing Oracle user as the Tivoli Workload Scheduler Oracle user, the installation process assumes that the configuration is complete and does not create the database objects for Tivoli Workload Scheduler. In this case, the installation completes successfully but you cannot use the database.

**Tivoli Workload Scheduler Oracle user password**

The password for the Tivoli Workload Scheduler Oracle user. It must comply with the Oracle naming rules.

**Create the Tivoli Workload Scheduler schema using the Oracle Partitioning option**

If you are installing on Oracle Enterprise Edition, you can choose to implement the Oracle Partitioning option to improve the performance of event-driven workload automation. For information about event-driven workload automation, see Overview.

**Tivoli Workload Scheduler data tablespace**

The name that identifies the Tivoli Workload Scheduler data table space. This table space must have been previously created by the database administrator. The default for this field is USERS.

**Tivoli Workload Scheduler reports tablespace**

The name that identifies the Tivoli Workload Scheduler table space where report data is to be stored. You can view the report data using the Dynamic Workload Console.

This table space must have been previously created by the database administrator. The default value for this field is USERS.

**Tivoli Workload Scheduler temporary tablespace**

The name that identifies the Tivoli Workload Scheduler temporary table space. This table space must have been previously created by the database administrator. The default value for this field is TEMP.

**Dynamic workload broker configuration information:**

Applies only to master domain manager installation.

**Dynamic workload broker workstation name**

The definition of the dynamic workload broker workstation created in the Tivoli Workload Scheduler database. Its type is broker. Spaces are not allowed and the maximum field length is 16 characters. It can contain alphanumeric, dash (-), and underscore (_) characters. The first character must be a letter.
Dynamic workload broker Netman port
The port on the workload broker workstation. The Tivoli Workload Scheduler master or backup master use this port to communicate with dynamic workload broker. The default value is 41114. The valid range is from 1 to 65535.

Installing a dynamic domain manager or backup dynamic domain manager
Install this component if you want to schedule and control your static and dynamic workload both in distributed and end-to-end environments. For example you have different branch offices and you want to run your dynamic schedule independently at each branch office to improve agent scalability. Moreover installing these components you run your dynamic schedule even if the master domain manager or the backup master domain manager is unavailable.

By installing a dynamic domain manager you can:
• Improve fault-tolerant and dynamic agents scalability because the workload of the agents in the domain is directly controlled by the dynamic domain manager to which they are directly connected.
• Allow static and dynamic processing to continue even if the agents connection to their master domain manager is unavailable.

If you want to ensure that your workload runs even if the connection to the dynamic domain manager is unavailable, install a backup dynamic domain manager.

A dynamic domain manager or a backup dynamic domain manager is composed of a:
• Fault-tolerant agent
• Broker server
• Dynamic agent
• Plan Connector

When you install a dynamic domain manager the following workstation types are created in the database:

Broker
For the broker server

Agent
For the dynamic agent

Manager
For the fault-tolerant agent

Domain
For the domain. The domain is a child of the master domain manager domain.

These workstations are created under the default master domain manager domain. If you want you can change it after the installation.

The fault-tolerant agent must be configured manually after you perform the installation:

If you are installing a dynamic domain manager:
Configure it as manager. See "Configuring a dynamic domain manager" on page 163.
If you are installing a backup dynamic domain manager:

Configure it as fta. See “Configuring a backup dynamic domain manager” on page 163.

For a graphical installation, from the installation DVD or from the appropriate elmage, start the launchpad as described in “Launchpad” on page 30 and select the Tivoli Workload Scheduler installation, or run the setup for the operating system on which you are installing:

On Windows operating systems

TWS\operating_system\SETUP.exe or TWS\SETUP.cmd

On UNIX and Linux operating systems

TWS/operating_system/SETUP.bin or TWS/SETUP.sh

Note: Your RDBMS must be running when you begin the installation.

If you are installing on UNIX or Linux and you want the installation to stop if the Tivoli Workload Scheduler prerequisite check encounters an error or warning, specify the parameter `-W checkPrerequisites.stopOnCheckPrereq=true` after the setup command. See “Checking prerequisites (UNIX and Linux)” on page 24 for more information about the prerequisite check.

The installation steps for the dynamic domain manager or the backup dynamic domain manager installation process are described in the following sections:

1. “Tivoli Workload Scheduler installation options”
2. “WebSphere Application Server installation options” on page 64
3. “RDBMS installation options” on page 65
   This section is divided into the following subsections:
   - “Installing for a DB2 database server” on page 65
   - “Installing for a DB2 database client” on page 67
   - “Installing for an Oracle database” on page 69
4. “Dynamic workload broker configuration information” on page 77

After a successful installation, perform one of the following configuration tasks, depending on the type of component you installed:

- “Configuring a dynamic domain manager” on page 163
- “Configuring a backup dynamic domain manager” on page 163.

Tivoli Workload Scheduler installation options:

Complete the following Tivoli Workload Scheduler installation options:

User name

Specify the Tivoli Workload Scheduler user name. User name can contain alphanumeric, dash (-), and underscore (_) characters. Cannot contain national characters. The first character of the user name must be a letter.

On Windows operating systems

If this user account does not already exist, it is automatically created by the installation wizard. 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. Create a user with a home directory and group. By default, Tivoli Workload Scheduler is installed in TWA_home.

**Note:** If you are installing into a new instance of Tivoli Workload Automation, the Tivoli Workload Scheduler user name and password are also used as the WebSphere Application Server administrator user name and password.

**Password**
Specify the Tivoli Workload Scheduler password. The password must comply with the password policy in your Local Security Settings. Spaces are not permitted.

**On Windows operating systems**
Password for users can include alphanumeric, dash (-), underscore (_) characters, and ()!?*=~/\ $`+;.:,.@.

**On other platforms**
Password for users can include any alphanumeric, dash (-), underscore (_) characters, and ()!?=+-..

**Dynamic domain manager and backup dynamic domain manager configuration information**

**Domain name**
Applies only to dynamic domain manager. Specify the Tivoli Workload Scheduler domain name managed by the dynamic domain manager. The default value is DYNAMICDM.

**This workstation name**
The name of the workstation where you are installing the instance. The default is the host name of the workstation. When you install:

- A dynamic domain manager, the name you specify here is the name of the Tivoli Workload Scheduler workstation known in the database as fta. Configure it as manager by performing the procedure described in "Configuring a dynamic domain manager" on page 163.
- A backup dynamic domain manager, the name you specify here is the name of the Tivoli Workload Scheduler workstation known in the database as fta. Configure it as fta by performing the procedure described in "Configuring a backup dynamic domain manager" on page 163.

Spaces are not allowed and the maximum field length is 16 characters. If the host name is longer than 16 characters, an alternative name must be provided for a successful installation. It can contain alphanumeric, dash (-), and underscore (_) characters. The first character must be a letter. The workstation name and master domain manager name cannot be the same.

**Master domain manager name**
The name of the master domain manager workstation. Spaces are not allowed and the maximum field length is 16 characters. The first character cannot be numeric. The workstation name and master domain manager name cannot be the same.

In case you are installing the dynamic domain manager to connect it only to the Tivoli Workload Scheduler for z/OS controller, the
value you must specify in this field is not used because in a z/OS lightweight end-to-end configuration the fault-tolerant agent is not needed.

**Tivoli Workload Scheduler Netman port**
The port used by the Netman process to run distributed scheduling. Netman is the network process that controls the production environment. The default value is **31111**. The valid range is from 1 to 65535.

In case you are installing the dynamic domain manager to connect it only to the Tivoli Workload Scheduler for z/OS controller, the value you must specify in this field is not used because in a z/OS lightweight end-to-end configuration the fault-tolerant agent is not needed.

**Note:** If you change this value, all default port number values in the application server port information panel are changed to reflect the new range. For example, if you specify 42111 as TCP/IP port Number, the default for HTTP transport becomes 42125, the default for HTTPS becomes 42126, and so on.

**Dynamic agent configuration information**

**Host name or IP address**
The local fully qualified host or IP address of the dynamic agent. The dynamic workload broker and the Tivoli Workload Scheduler for z/OS controller use this address to connect to the dynamic agent.

**Agent display name**
The name of the dynamic agent workstation definition.

**JobManager port number**
The dynamic agent secure port number (SECUREADDR). The Tivoli Workload Scheduler for z/OS controller and the dynamic workload broker use this port to connect to the Tivoli Workload Scheduler dynamic agent. The default value is **31114**. The valid range is from 1 to 65535.

**Enable HTTPS communication for the JobManager port**
This option enables HTTPS communication between the local dynamic workload broker and the dynamic agent. For secure connections, it is recommended that you use HTTPS. To use HTTP communication, clear this checkbox.

**Dynamic workload broker host name**
Applies only to backup dynamic domain manager. The fully qualified host name of the dynamic workload broker. Tivoli Workload Scheduler for z/OS controller and the dynamic agent use this address to contact the dynamic workload broker.

**Dynamic workload broker HTTPS port number**
Applies only to backup dynamic domain manager. The HTTPS port of the dynamic workload broker. You specified it when installing the dynamic domain manager. The dynamic agent uses it to connect to the dynamic workload broker. The default value is **31116**. If you leave the field blank, it defaults to 0. The valid range is from 1 to 65535.
Automatically generate all the other Tivoli Workload Scheduler ports
If you check this box, all the ports needed by WebSphere Application Server are automatically generated using the default values and the application server port information panel is not displayed. The installation procedure checks for the availability of the ports in the specified port range. If one or more ports are in use by other applications, you are prompted to enter a new port number. By default, this box is checked. If you have not requested to generate ports automatically, specify the values for the ports used by the application server embedded in the Tivoli Workload Scheduler instance. Accept the default values unless you know that they are already in use by other applications.

Installation directory
Enter the name of the directory where the dynamic domain manager or backup dynamic domain manager instance is installed for the specified user. The maximum field length is 46 characters and the name must not contain numbers. Parentheses () are not allowed. You cannot use national characters.

Spaces are allowed. However, you cannot install Tivoli Workload Scheduler for Applications version 8.2.1 or earlier on the current version of Tivoli Workload Scheduler if the directory path contains spaces.

On Windows operating systems
• The name must be longer than three characters, the second character must be ;, and the third character must be \.
• The default directory is the default directory is %ProgramFiles%\IBM\TWA.

On UNIX and Linux operating systems
• The name must be longer than one character and the first character must be /.
• The default directory is the /opt/IBM/TWA directory.
• Optionally check Create symbolic links to create links in the /usr/bin directory. Any existing Tivoli Workload Scheduler symbolic links are overwritten.

WebSphere Application Server installation options:
The following installation options are provided for WebSphere Application Server data. The installation options you complete depend upon whether you are installing into a new instance of Tivoli Workload Automation or into an existing instance.

The installation procedure checks for the availability of the ports in the specified port range. If one or more ports are in use by other applications, you are prompted to enter a new port number.

New instance
If you are installing into a new instance of Tivoli Workload Automation, provide the following information:

HTTP transport
The port for the HTTP transport. It is used by the composer command line interface when this protocol is selected. The default value is 31115. The valid range is from 1 to 65535.

HTTPS transport
The port for the secure HTTP transport. It is used by the composer
command line interface when this protocol is selected. The default value is 31116. The valid range is from 1 to 65535.

**Bootstrap**
- The port for the bootstrap or RMI. It is used by the graphical user interfaces. The default value is 31117. The valid range is from 1 to 65535.

**SOAP connector**
- The port for the application server protocol SOAP connector. The default value is 31118. The valid range is from 1 to 65535.

**SAS Server Authentication Listener**
- The port used by the Secure Association Services (SAS) to listen for inbound authentication requests. The default value is 31119. The valid range is from 1 to 65535.

**CSIv2 Server Authentication Listener**
- The port on which the Common Secure Interoperability Version 2 (CSIv2) service listens for inbound server authentication requests. The default value is 31120. The valid range is from 1 to 65535.

**CSIv2 Client Authentication Listener**
- The port on which the Common Secure Interoperability Version 2 (CSIv2) service listens for inbound client authentication requests. The default value is 31121. The valid range is from 1 to 65535.

**ORB Listener**
- The port used for RMI over IIOP communication. The default value is 31122. The valid range is from 1 to 65535.

**Administration HTTP transport**
- The administrative console port. The default value is 31123. The valid range is from 1 to 65535.

**Administration HTTPS transport**
- The administrative console secure port. The default value is 31124. The valid range is from 1 to 65535.

**Existing instance**
- If you are installing into an existing instance of Tivoli Workload Automation, you are only required to provide the WebSphere Application Server administrator user name and password. The port data is retrieved automatically from the existing WebSphere Application Server instance. If you do not know the user name, click Retrieve on the appropriate panel to populate this field, although you must still provide the password. If you click Retrieve, you might have to wait for the field to populate.

**RDBMS steps:**
This section is divided into subsections. See the section that corresponds to the RDBMS you are using.

- “Installing for a DB2 database server” on page 65
- “Installing for a DB2 database client” on page 67
- “Installing for an Oracle database” on page 69

**Note:** When providing the database name ensure to provide the name of a database that is not used by a master domain manager.

**Dynamic workload broker configuration information:**
Applies only to dynamic domain manager installation.
Dynamic workload broker workstation name
The definition of the dynamic workload broker workstation created in the
Tivoli Workload Scheduler database. Spaces are not allowed and the
maximum field length is 16 characters. It can contain alphanumeric, dash
(-), and underscore (_) characters. The first character must be a letter.

The dynamic workload broker workstation acts as the communication
bridge between the master domain manager and the dynamic workload
broker component. In your job or job stream definitions, it is the
workstation where the jobs run. In this way, you submit your workload
through this workstation to the dynamic workload broker component.

Dynamic workload broker Netman port
The port on the workload broker workstation used by the Tivoli Workload
Scheduler dynamic domain manager or backup dynamic domain manager
to communicate with dynamic workload broker. The default value is 41114.
The valid range is from 1 to 65535.

Do you want to connect the Dynamic Domain Manager only to the z/OS
controller?
Select this check box if you want to connect the dynamic domain manager
only to the z/OS controller. Leave the check box clear to connect the
dynamic domain manager to:
• A master domain manager
• Both a master domain manager and a z/z/OS controller

When you connect the Dynamic Domain Manager only to a z/OS
controller, is because you want to create a lightweight end-to-end
scheduling environment in which to schedule and control workload from
the Tivoli Workload Scheduler for z/OS on distributed systems. To
complete this environment you must install a Tivoli Workload Scheduler
for z/OS agent. See the Tivoli Workload Scheduler for z/OS: Planning and
Installation Guide for a detailed explanation on how to install the Tivoli
Workload Scheduler for z/OS agent. When you select the check box the
Master domain manager host name and the Master domain manager
https port field are disabled.

Master domain manager host name
The fully qualified host name on which the dynamic domain manager
contacts the master domain manager.

Master domain manager https port
The port for the secure HTTP transport. It is used by the dynamic domain
manager to contact the master domain manager. The default value is 31116.
The valid range is from 1 to 65535.

Installing an agent
This section describes how to install the Tivoli Workload Scheduler fault-tolerant
and dynamic agents in your distributed or end-to-end network. During the
installation of the dynamic agent, you can add the Java runtime environment to
run job types with advanced options, both the job types supplied with the product
and the additional types implemented through the custom plug-ins. To install the
Tivoli Workload Scheduler for z/OS connector, see Tivoli Workload Scheduler for
z/OS: Planning and Installation Guide.

Note: IBM i® is an exception. The dynamic agent on this platform can be installed
only using the twsinst command line.
To install from the DVD or eImages, run the setup for the operating system on which you are installing as follows:

**On Windows operating systems:**
TWS\operating_system\SETUP.exe or TWS\SETUP.cmd

**On UNIX and Linux operating systems:**
TWS/operating_system/SETUP.bin or TWS/SETUP.sh

**Note:** SETUP.sh copies the entire image to a temporary directory. Ensure that there is enough space available.

If you are installing on UNIX or Linux and you want the installation to stop if the Tivoli Workload Scheduler prerequisite check encounters an error or warning, specify the parameter `-W checkPrerequisites.stopOnCheckPrereq=true` after the setup command. See "Checking prerequisites (UNIX and Linux)” on page 24 for more information about the prerequisite check.

For a graphical installation, from the installation DVD or from the appropriate eImage, start the launchpad as described in "Launchpad" on page 30 and select the Tivoli Workload Scheduler installation.

**Windows operating systems**
- To install for a local user, you must be an administrator on the computer.
- To install for a domain user, you must be a domain administrator.

**UNIX or Linux operating systems**
To install on UNIX or Linux, you must be logged in as root.

You can install agents during the full installation of Tivoli Workload Scheduler or in a stand-alone installation of only the agent. You can optionally simultaneously install a fault-tolerant agent, a dynamic agent, and the Java runtime environment.

**Installing a fault-tolerant agent**
To install only a fault-tolerant agent, during the installation wizard you select agent as the component you want to install, and then select fault-tolerant as the type of agent. Ensure you deselect dynamic agent if you do not want to install that type of agent at the same time. After you performed the installation configure it manually. See "Configuring a fault-tolerant agent” on page 164.

**Installing a dynamic agent and adding Java runtime**
To install only a dynamic agent, during the installation wizard you select agent as the component you want to install. The check box for dynamic agent is already selected by default. Select Java runtime if you want to install the runtime environment to run job types with advanced options, both those types supplied with the product and the additional types implemented through the custom plug-ins. After you performed the installation, perform the steps outlined in "Configuring a dynamic agent” on page 165.

**Simultaneously installing both a fault-tolerant and a dynamic agent**
To install both a fault-tolerant agent and a dynamic agent, during the installation wizard you select agent as the component you want to install, and then select both fault-tolerant and dynamic agents (the check box for dynamic is already selected by default). Select Java runtime if you want to install the runtime environment to run job types with advanced options, both those types supplied with the product and the additional types
implemented through the custom plug-ins. Perform the steps outlined in “Configuring a dynamic agent” on page 165 and “Configuring a fault-tolerant agent” on page 164.

See “Installation options” for descriptions of the installation parameters and options.

Installation options:
This section describes the installation configuration parameters and options.

User name
Specify the Tivoli Workload Scheduler user name. Spaces are not permitted.

On Windows operating systems
- If this user account does not already exist, it is automatically created by the installation wizard.
- If you specify a domain user specify the name as domain_name\user_name.
- If you are installing in a domain controller the user name must always be domain_name\user_name.
- If you specify a local user with the same name as a domain user, the local user must first be created manually by an administrator and then specified as system_name\user_name.

On UNIX and Linux operating systems
This user account must be created manually before running the installation. Create a user with a home directory. By default, Tivoli Workload Scheduler is installed under the HOME directory of the selected user.

Note: If you are installing an agent in an instance of Tivoli Workload Automation where the Dynamic Workload Console is already installed, and you intend later to add the Connector feature to the agent, you are strongly advised to use, as the TWS_user of the agent, the same user that you used as the WebSphere Application Server administration user when you installed the Dynamic Workload Console. The WebSphere Application Server administration is simplified if the names are the same (this is the default if you install a Tivoli Workload Scheduler component that uses WebSphere Application Server before the Dynamic Workload Console).

Password
Specify the Tivoli Workload Scheduler password. The password must comply with the password policy in your Local Security Settings. Spaces are not permitted.

On Windows operating systems
Password for users can include alphanumerics, dash (-), underscore (_) characters, and ()!?*=^~/[]$`+;,.@.

On other platforms
Password for users can include any alphanumeric, dash (-), underscore (_) characters, and ()!?*=^~/..
Company
The name of the company. Spaces are allowed and the maximum field length is 40 characters.

This workstation name
The name of the workstation where you are installing the instance. The default is the host name of the workstation. The name you specify here is the name of the Tivoli Workload Scheduler workstation as it is known in the database. The name must start with a letter, and can contain alphanumeric characters, dashes, and underscores. It can contain up to 16 characters. If the hostname is longer than 16 characters, an alternative name must be provided.

Master domain manager name
The name of the master domain manager to which the workstation belongs.

Tivoli Workload Scheduler Netman port
The port 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.

Dynamic agent configuration information

Host name or IP address
The fully qualified host name or IP address of the dynamic agent. The dynamic workload broker uses this address to connect to the dynamic agent.

Agent display name
The name of the dynamic agent workstation definition

JobManager port number
The dynamic agent port (SECUREADDR or TCPADDR) of the dynamic agent. Dynamic workload broker uses this port to connect to the dynamic workload broker. It is used by JobManager to run dynamic workload and to run workload coming from a z/OS environment in a distributed environment. JobManager is the network process that controls the dynamic scheduling environment and the z-centric environment. The installation default value is 31114. The valid range is from 1 to 65535.

Enable HTTPS communication for the JobManager port
This option enables HTTPS communication between the Tivoli Workload Scheduler master domain manager or Tivoli Workload Scheduler for z/OS controller and the agent. If you accept this default, ensure that you also configure HTTPS communication on the z/OS controller. For secure connections, it is recommended that you use HTTPS. To use HTTP communication, clear this checkbox. However, to improved performance of communication between the Tivoli Workload Scheduler for z/OS controller and the agent, choose to use HTTP.

Dynamic workload broker host name
The fully qualified host name of the dynamic workload broker. The Tivoli Workload Scheduler dynamic agent uses it to connect to the dynamic workload broker.

Dynamic workload broker HTTPS port number
The HTTPS transport port of the dynamic workload broker. You specified it when installing the master or backup master. The
dynamic agent uses this port to connect to the dynamic workload broker. The installation default value is 31116 although if you leave the field blank, it defaults to 0. The valid range is from 1 to 65535.

**Installation directory**

Enter the name of the directory where the Tivoli Workload Scheduler instance will be installed for the specified user. The maximum field length is 46 characters and the name must not contain numbers, parentheses () or national characters.

Spaces are allowed however, Tivoli Workload Scheduler for Applications version 8.2.1 or earlier cannot be installed on the current version of Tivoli Workload Scheduler if the directory path contains spaces.

**On Windows operating systems**

- The name must be longer than three characters, the second character must be :, and the third character must be \.
- The default directory is %ProgramFiles%\IBM\TWA.

**On UNIX and Linux systems**

- The name must be longer than one character and the first character must be /.
- The default directory is the /opt/IBM/TWA directory.
- Optionally check **Create symbolic links** to create links in the /usr/bin directory. Any existing Tivoli Workload Scheduler symbolic links are overwritten.

**Installing a command line client**

The command line client is a component of Tivoli Workload Scheduler that implements many of the commands used on the master domain manager. It can be installed on any workstation that satisfies its prerequisites, including workstations where no other Tivoli Workload Scheduler components are installed. It communicates by TCP/IP with the command line server, which is part of the master domain manager. Install the command line client using the installation wizard or in silent mode. Do not install the command-line client in the same path that you used to install any other Tivoli Workload Scheduler component.

The information required to make the connection with the master domain manager must be defined either in the local options file or supplied as parameters to the command.

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)
- sendevent

**Note:** The command line client is different from and independent from the ability to use **conman** locally on an agent to manage the local Symphony™ file and local jobs. Configuring a connection with the master does allow the local **conman** to submit objects from the database into the plan.

**Running the installation wizard:**

To install a command line client on an existing installation, perform the following steps:
1. For a graphical installation, from the installation DVD or from the appropriate
eImage, start the launchpad as described in “Launchpad” on page 30 and select
the Tivoli Workload Scheduler installation, or run the setup for the operating
system on which you are installing.

From the TWS directory on the DVD, perform the following:

- On Windows: WINDOWS\SETUP.exe or SETUP.cmd
- On UNIX and Linux: SETUP.sh or operating_system/SETUP.bin.

Note: SETUP.sh copies the entire image to a temporary directory. Ensure there
is enough space available.

2. Follow the installation wizard screens to complete the installation. The
following list describes the fields that you might need to complete during the
installation.

Remote Host
   The TCP/IP address or host name of the workstation where the Tivoli
   Workload Scheduler engine is installed.

Remote Port
   The HTTP or HTTPS port number used to connect to the workstation
   where the master domain manager is installed. This port number must
   match the values defined for the master domain manager.

   Note: The default protocol used by the command line client to
   establish a connection with the master is https. If the http port is
   specified in the Remote Port field, before running any
   commands, you must modify the PROTOCOL property in the
   localopts file by inserting http instead of https.

User Name
   The user name used to connect to the workstation where the master
   domain manager is installed. This user should be a valid user listed in
   the security file on the master domain manager.

Password
   The password used to connect to the workstation where the master
   domain manager is installed.

Adding a feature
To some of the components of the product you can add a feature:

Add a connector to an agent
   If you want to access the plan which is available on an agent or domain
   manager workstation, add the connector feature which enables the
   communication between it and the Dynamic Workload Console. See
   “Adding a connector” on page 166.

Add a language pack to the command-line client
   If you want to use the command-line client in a supported language other
   than English or the language of the locale of the computer where you
   installed it, add the language pack. See “Adding language packs to a
   command line client” on page 167.

Add the Java runtime to an agent
   During the installation of the agent you might have chosen not to add the
   Java runtime that supports the running of job types with advanced
   options. If you later decide that you require this function, you can just add
   the Java runtime separately. See “Adding the Java runtime” on page 102.
If you already set up your environment and want to enable dynamic scheduling capabilities, see “Enabling dynamic scheduling after installation” on page 168.

**Performing a silent installation**

A silent installation runs according to parameters set in a response file. The response file includes all the installation information required to run the installation without user intervention.

There are two ways to customize a response file to satisfy your installation requirements:

- Edit an existing response file template provided on the installation DVDs. See “Silent installation using response file templates” on page 87.
- Automatically create a customized response file by running the installation wizard. See “Silent installation using an automatically generated response file” on page 88.

Table 6 lists the response files and the types of installation each performs by platform:

<table>
<thead>
<tr>
<th>Type of installation</th>
<th>Response file to use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installing on UNIX</td>
<td></td>
</tr>
<tr>
<td>Command line client with language packs</td>
<td>TWS86_CLI_LP_UNIX.txt</td>
</tr>
<tr>
<td>Command line client (no language packs installed)</td>
<td>TWS86_CLI_UNIX.txt</td>
</tr>
<tr>
<td>Fresh fault-tolerant agent or both fault-tolerant agent and dynamic agent on existing Tivoli Workload Automation instance</td>
<td>TWS86_FRESH_Agent_existTWA_UNIX.txt</td>
</tr>
<tr>
<td>Fresh fault-tolerant agent or both fault-tolerant agent and dynamic agent on new Tivoli Workload Automation instance</td>
<td>TWS86_FRESH_Agent_newTWA_UNIX.txt</td>
</tr>
<tr>
<td>Fresh backup dynamic domain manager on existing Tivoli Workload Automation instance</td>
<td>TWS86_FRESH_BACKUP_DDM_existTWA_UNIX.txt</td>
</tr>
<tr>
<td>Fresh backup dynamic domain manager on new Tivoli Workload Automation instance</td>
<td>TWS86_FRESH_BACKUP_DDM_newTWA_UNIX.txt</td>
</tr>
<tr>
<td>Fresh backup master domain manager on existing Tivoli Workload Automation instance</td>
<td>TWS86_FRESH_BACKUP_MDM_existTWA_UNIX.txt</td>
</tr>
<tr>
<td>Fresh backup master domain manager on new Tivoli Workload Automation instance</td>
<td>TWS86_FRESH_BACKUP_MDM_newTWA_UNIX.txt</td>
</tr>
<tr>
<td>Fresh connector with no Dynamic Workload Console installed</td>
<td>TWS86_FRESH_Conn_NEITHER_TDWC_NOR_ZCONN_UNIX.txt</td>
</tr>
<tr>
<td>Fresh connector on the Dynamic Workload Console</td>
<td>TWS86_FRESH_Conn_ON_TDWC_OR_ZCONN_UNIX.txt</td>
</tr>
<tr>
<td>Fresh dynamic agent</td>
<td>TWS86_FRESH_DYNAMIC_Agent_UNIX.txt</td>
</tr>
<tr>
<td>Type of installation</td>
<td>Response file to use</td>
</tr>
<tr>
<td>--------------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td>Fresh dynamic domain manager on existing Tivoli Workload Automation instance for Tivoli Workload Scheduler for z/OS lightweight end-to-end scheduling environment</td>
<td>TWS86_FRESH_DDM_for_ZOS_existTWA_UNIX.txt</td>
</tr>
<tr>
<td>Fresh dynamic domain manager on existing Tivoli Workload Automation instance</td>
<td>TWS86_FRESH_DDM_existTWA_UNIX.txt</td>
</tr>
<tr>
<td>Fresh dynamic agent</td>
<td>TWS86_FRESH_DYNAMIC_Agent_UNIX.txt</td>
</tr>
<tr>
<td>Fresh dynamic domain manager on new Tivoli Workload Automation instance for Tivoli Workload Scheduler for z/OS lightweight end-to-end scheduling environment</td>
<td>TWS86_FRESH_DDM_for_ZOS_newTWA_UNIX.txt</td>
</tr>
<tr>
<td>Fresh dynamic domain manager on new Tivoli Workload Automation instance</td>
<td>TWS86_FRESH_DDM_newTWA_UNIX.txt</td>
</tr>
<tr>
<td>Fresh master domain manager on existing Tivoli Workload Automation instance</td>
<td>TWS86_FRESH_MDM_existTWA_UNIX.txt</td>
</tr>
<tr>
<td>Fresh master domain manager on new Tivoli Workload Automation instance</td>
<td>TWS86_FRESH_MDM_newTWA_UNIX.txt</td>
</tr>
<tr>
<td>Uninstall an agent</td>
<td>TWS86_UNINSTALL_Agent.txt</td>
</tr>
<tr>
<td>Upgrade an agent</td>
<td>TWS86_UPGRADE_Agent_UNIX.txt</td>
</tr>
<tr>
<td>Upgrade a backup master domain manager from version 8.3 or later</td>
<td>TWS86_UPGRADE_BACKUP_MDM_83plus_UNIX.txt</td>
</tr>
<tr>
<td>Upgrade a command line client</td>
<td>TWS86_UPGRADE_CLI_UNIX.txt</td>
</tr>
<tr>
<td>Upgrade a connector on an end-to-end fault-tolerant agent</td>
<td>TWS86_UPGRADE_Connector_and_FTA_UNIX.txt</td>
</tr>
<tr>
<td>Upgrade a master domain manager from version 8.3 or later</td>
<td>TWS86_UPGRADE_MDM_83plus_UNIX.txt</td>
</tr>
</tbody>
</table>

**Installing on Windows**

<table>
<thead>
<tr>
<th>Type of installation</th>
<th>Response file to use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command line client with language packs</td>
<td>TWS86_CLI_LP_WIN.txt</td>
</tr>
<tr>
<td>Command line client (no language packs installed)</td>
<td>TWS86_CLI_WIN.txt</td>
</tr>
<tr>
<td>Fresh fault-tolerant agent or both fault-tolerant agent and dynamic agent on existing Tivoli Workload Automation instance</td>
<td>TWS86_FRESH_Agent_existTWA_WIN.txt</td>
</tr>
<tr>
<td>Fresh fault-tolerant agent or both fault-tolerant agent and dynamic agent on new Tivoli Workload Automation instance</td>
<td>TWS86_FRESH_Agent_newTWA_WIN.txt</td>
</tr>
<tr>
<td>Type of installation</td>
<td>Response file to use</td>
</tr>
<tr>
<td>--------------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td>Fresh backup master domain manager on existing Tivoli Workload Automation instance</td>
<td>TWS86_FRESH_BACKUP_MDM_existTWA_WIN.txt</td>
</tr>
<tr>
<td>Fresh backup dynamic domain manager on new Tivoli Workload Automation instance</td>
<td>TWS86_FRESH_BACKUP_DDM_newTWA_WIN.txt</td>
</tr>
<tr>
<td>Fresh backup dynamic domain manager on existing Tivoli Workload Automation instance</td>
<td>TWS86_FRESH_BACKUP_DDM_existTWA_WIN.txt</td>
</tr>
<tr>
<td>Fresh connector with no Dynamic Workload Console installed</td>
<td>TWS86_FRESH_Conn_NEITHER_TDWC_NOR_ZCONN_WIN.txt</td>
</tr>
<tr>
<td>Fresh connector on the Dynamic Workload Console</td>
<td>TWS86_FRESH_Conn_ON_TDWC_OR_ZCONN_WIN.txt</td>
</tr>
<tr>
<td>Fresh dynamic agent</td>
<td>TWS86_FRESH_DYNAMIC_Agent_WIN.txt</td>
</tr>
<tr>
<td>Fresh dynamic domain manager on existing Tivoli Workload Automation instance for Tivoli Workload Scheduler for z/OS lightweight end-to-end scheduling environment</td>
<td>TWS86_FRESH_DDM_existTWA_WIN.txt</td>
</tr>
<tr>
<td>Fresh dynamic domain manager on existing Tivoli Workload Automation instance</td>
<td>TWS86_FRESH_DDM_for_ZOS_existTWA_WIN.txt</td>
</tr>
<tr>
<td>Fresh dynamic domain manager on new Tivoli Workload Automation instance for Tivoli Workload Scheduler for z/OS lightweight end-to-end scheduling environment</td>
<td>TWS86_FRESH_DDM_for_ZOS_newTWA_WIN.txt</td>
</tr>
<tr>
<td>Fresh dynamic domain manager on new Tivoli Workload Automation instance</td>
<td>TWS86_FRESH_DDM_newTWA_WIN.txt</td>
</tr>
<tr>
<td>Fresh master domain manager on existing Tivoli Workload Automation instance</td>
<td>TWS86_FRESH_MDM_existTWA_WIN.txt</td>
</tr>
<tr>
<td>Fresh master domain manager on new Tivoli Workload Automation instance</td>
<td>TWS86_FRESH_MDM_newTWA_WIN.txt</td>
</tr>
<tr>
<td>Uninstall an agent</td>
<td>TWS86_UNINSTALL_Agent.txt</td>
</tr>
<tr>
<td>Upgrade an agent</td>
<td>TWS86_UPGRADE_Agent_WIN.txt</td>
</tr>
<tr>
<td>Upgrade a command line client</td>
<td>TWS86_UPGRADE_CLI_WIN.txt</td>
</tr>
<tr>
<td>Upgrade a backup master domain manager from version 8.3 or later</td>
<td>TWS86_UPGRADE_BACKUP_MDM_83plus_WIN.txt</td>
</tr>
<tr>
<td>Upgrade a command line client</td>
<td>TWS86_UPGRADE_CLI_WIN.txt</td>
</tr>
<tr>
<td>Upgrade a connector</td>
<td>TWS86_UPGRADE_Connector_WIN.txt</td>
</tr>
<tr>
<td>Upgrade a connector on an end-to-end fault-tolerant agent</td>
<td>TWS86_UPGRADE_Connector_and_FTA_WIN.txt</td>
</tr>
<tr>
<td>Upgrade a master domain manager from version 8.3 or later</td>
<td>TWS86_UPGRADE_MDM_83plus_WIN.txt</td>
</tr>
</tbody>
</table>
Note: When you are performing a silent installation on UNIX zSeries® systems, you must first save the response file in UTF 8 format.

To perform a silent installation using a response file template, perform the following steps:

1. Copy the relevant response file to a local directory and edit it to meet the needs of your environment.

   **Note:** Be sure to review the license agreement information included in the installation media. To accept the terms of the license agreement, set the `licenseAccepted` parameter to `true` in the response file you are using. This value is required to complete the silent installation successfully.

2. Save the file with your changes.

3. Enter the following command:

   **Windows operating systems**
   ```
   SETUP.exe -options <local_dir>\response_file.txt -silent
   ```
   Where `response_file.txt` is the name of the response file to be used for installation. The `SETUP.exe` file is located in the `WINDOWS` directory. See “Installation media” on page 26.

   **UNIX and Linux operating systems**
   ```
   ./SETUP.bin -options <local_dir>/response_file.txt -silent
   ```
   Where `response_file.txt` is the name of the response file to be used for installation. The `SETUP.sh` file is located in the root directory of the relevant installation DVD. See “Installation media” on page 26.

4. Review the installation messages in the `summary.log` file to check that installation was successful.

5. At the end of a successful installation, perform one of the following configuration tasks, depending on the type of agent you installed:
   - "Configuring a master domain manager" on page 160
   - "Configuring a fault-tolerant agent" on page 164
   - "Configuring a dynamic agent" on page 165

   **Note:** If you are installing an agent in an instance of Tivoli Workload Automation where the Dynamic Workload Console is already installed, and you intend later to add the Connector feature to the agent, you are strongly advised to use, as the `<TWS_user>` of the agent, the same user as you used as the WebSphere Application Server administration user when you installed the Dynamic Workload Console. It will make the WebSphere Application Server administration easier to perform if the names are the same (this is the default case if you install a Tivoli Workload Scheduler component that uses WebSphere Application Server before the Dynamic Workload Console).

**Silent installation using response file templates**
Edit the response file templates provided on the installation DVDs in the `\TWS\RESPONSEFILES\` directory. Instructions for customizing the files are included in the files as commented text. For details about response file properties, see “The Tivoli Workload Scheduler response file properties” on page 292.
Silent installation using an automatically generated response file

During the initial installation of the current version of Tivoli Workload Scheduler, you can create a response file based on the parameters of the initial installation. You use this response file to run subsequent installations with the same parameters. Creating an automatically generated response file is recommended because all input is automatically validated by the program.

To perform a silent installation using an automatically generated response file, perform the following steps:

1. Perform the initial installation using the following command:

   **Windows operating systems**
   
   ```
   SETUP.exe -options-record <local_dir>\response_file.txt
   ```
   
   Where `response_file.txt` is the name of the response file to be created. The SETUP.exe file is located in the WINDOWS directory. See “Installation media” on page 26.

   **UNIX and Linux operating systems**
   
   ```
   ./SETUP.bin -options-record <local_dir>/response_file.txt
   ```
   
   Where `response_file.txt` is the name of the response file to be created. The SETUP.sh file is located in the root directory of the relevant installation DVD. See “Installation media” on page 26.

2. For all subsequent installations, enter the following command:

   ```
   | Windows     | SETUP.exe -options <local_dir>\response_file.txt -silent |
   | UNIX and Linux | ./SETUP.bin -options <local_dir>/response_file.txt -silent |
   ```

3. After each silent installation, review the installation messages in the `summary.log` file to check that installation was successful.

4. At the end of a successful installation, perform one of the following configuration tasks, depending on the type of agent you installed:

   - “Configuring a master domain manager” on page 160
   - “Configuring a fault-tolerant agent” on page 164
   - “Configuring a dynamic agent” on page 165

Installing agents using twsinst

This section describes how to use the `twsinst` script to install the Tivoli Workload Scheduler fault-tolerant or dynamic agent in your distributed or end-to-end network. The `twsinst` script is an alternative to the silent installation wizard (see “Performing a silent installation” on page 84). In addition, if you are installing the dynamic agent, you can use the `twsinst` script to add to the agent the Java runtime necessary to run job types with advanced options. Agents installed using `twsinst` can only be uninstalled using `twsinst`.

**Note:**

1. The dynamic agent on IBM i® can be installed only using the `twsinst` command line.
2. You cannot use the `twsinst` script to install Tivoli Workload Scheduler agents if you are running a Java Virtual Machine (JVM).
During the installation process, `twsinst` creates a file in the following directories for each of the installation steps:

**On UNIX operating systems**

```
/user's_home/TWS
```

**On Windows operating systems**

```
%ProgramFiles%\IBM\TWA\TWS
```

### Installing the agents

You can install the fault-tolerant or dynamic agent during the full installation of Tivoli Workload Scheduler or in a stand-alone installation of just the agent.

See "Agent installation parameters" on page 92 for descriptions of the installation parameters and options.

To install a Tivoli Workload Scheduler agent, perform the following steps:

**On UNIX and Linux operating systems:**

1. Insert the DVD for your operating system or download the agent image (see "Installation media" on page 26 or the Download Document at http://www.ibm.com/support/docview.wss?rs=672&uid=swg24027501 for more information).
2. Create the Tivoli Workload Scheduler 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 Tivoli Workload Scheduler user.)

3. Log in as root on the workstation where you want to install the product.
4. From the `DVD_root/TWS/operating_system` directory, run `twsinst` using the syntax described below. See "Agent installation parameters" on page 92 for descriptions of the syntax parameters.

**On Windows operating systems:**

1. Insert the DVD for your operating system or download the agent image (see "Installation media" on page 26 or the Download Document at http://www.ibm.com/support/docview.wss?rs=672&uid=swg24027501 for more information).
2. Log in as administrator on the workstation where you want to install the product.
3. From the `DVD_root/TWS/operating_system` directory, run `twsinst` using the syntax described below. See "Agent installation parameters" on page 92 for descriptions of the syntax parameters.

**Note:** `twsinst` for Windows is a Visual Basic Script (VBS) that you can run in CScript and WScript mode. The Tivoli Workload Scheduler user is automatically created. The software is installed by default in the Tivoli Workload Scheduler installation directory. The default value is `%ProgramFiles%\IBM\TWA`.

**On IBM i operating systems, dynamic agent only**

1. On the IBM i workstation, sign on as `QSECOFR` user.
2. Create the IBM i user profile for which the Tivoli Workload Scheduler dynamic agent is installed. This user profile is not to be confused with the user performing the installation logged on as QSECOFR, but instead is 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 page 92. 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, QDOCS, QLPAUTO, QLPINSTALL, QRJE, QSECOFR, QSQL, QSYS, QTSTRQS

Important:

- If the user profile is member of a group, the installation fails. Set the group profile associated with the user profile to *NONE.
- On IBM i, if username is longer than 8 characters, after installation the agent (and the JobManager component) will run under the QSECOFR user, while they should run 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 dynamic agent user.

4. Insert the DVD for the IBM i system or download the agent eImage from the Passport Advantage Online website (see “Installation media” on page 26 or the Download Document at http://www.ibm.com/support/docview.wss?rs=672&uid=swg24027501 for more information).

5. If you have download the eImages, to untar the package, you can use the PASE shell or the AIXterm.

Using PASEshell:

a. Open the PASE shell.
b. Run the command "CALL QP2TERM".
c. Locate the folder where you download the eImages and run the command:
"tar xvf TWS86_IBM_I.tar"
d. Exit from the PASE shell.

Using AIXterm:

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. Using this pop-up window, untar the file TWS86_IBM_I.tar.

6. Open a QSH shell and run the twsinst script. During the installation the product creates an IBM i library and a job description with the same name as the user profile created in Step 2. 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.
Note: If you do not run `twsinst` script from a `QSH shell` the installation fails.

A failed installation issues the return code RC = 1. In the case of a failed installation, see to the *Tivoli Workload Automation: Messages and Codes*.

At the end of 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 164
- "Configuring a dynamic agent" on page 165

On UNIX, Linux, and IBM i operating systems:

Show command usage and version

```
twsinst -u | -v
```

Install a new instance

```
twsinst -new -uname username
  [-addjruntime true|false]  
  [-agent fta|both]  
  [-company company_name]  
  [-create_link]  
  [-displayname agentname]  
  [-hostname hostname]  
  [-inst_dir install_dir]  
  [-jimport port_number]  
  [-jimportssl true|false]  
  [-lang lang_id]  
  [-master master_cpu_name]  
  [-port port_number]  
  [-reset_perm]  
  [-skip_usercheck]  
  [-stoponcheckprereq]  
  [-tdwbport tdwbport_number]  
  [-tdwbhostname host_name]  
  [-thiscpu workstation]
```

On Windows operating systems:

Show command usage and version

```
twsinst -u | -v
```

Install a new instance

```
twsinst -new -uname username
  -password user_password
  [-addjruntime true|false]  
  [-agent fta|both]  
  [-company company_name]  
  [-displayname agentname]  
  [-domain user_domain]  
  [-hostname hostname]  
  [-inst_dir install_dir]  
  [-jimport port_number]  
  [-jimportssl true|false]  
  [-lang lang_id]  
  [-master master_cpu_name]  
  [-port port_number]  
  [-skip_usercheck]  
  [-tdwbport tdwbport_number]  
  [-tdwbhostname host_name]  
  [-thiscpu workstation]
```
Agent installation parameters
This section lists and describes the parameters used when using the twsinst script to install the fault-tolerant or dynamic agent.

-addjruntime true | false
Adds the Java runtime to run job types with advanced options, both those types supplied with the product and the additional types implemented through the custom plug-ins. Valid values are true and false. The default for a fresh installation is true.

-agent dynamic | fta | both
The type of agent you want to install. Valid values are:

dynamic
To install the Tivoli Workload Scheduler dynamic agent. Use this value with the -tdwbhostname host_name and the -tdwbport tdlwpport_number parameters.

fta
To install the Tivoli Workload Scheduler fault-tolerant agent.

both
To install the dynamic agent used with the -tdwbhostname host_name and the -tdwbport tdlwpport_number parameters, and the fault-tolerant agent.

The default is dynamic.

-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 only. Create the symlink between /usr/bin/at and <install_dir>/TWS/bin/at. See Table 2 on page 26 for more information.

displayname
The name to assign to the dynamic agent. The default is the host name of this computer.

domain user_domain
Windows only. The domain name of the Tivoli Workload Scheduler user. The default is the name of the workstation on which you are installing the product.

-hostname host_name
The fully qualified host name or IP address on which the agent is contacted by the dynamic workload broker. The default is the host name of this computer.

-inst_dir installation_dir
The directory of the Tivoli Workload Scheduler installation.

On UNIX and Linux operating systems:
The path cannot contain blanks. If you do not manually specify a path, the path is set to the default home directory, that is, %ProgramFiles%\IBM\TWA.

On Windows operating systems:
If you specify a path that contains blanks, enclose it in double quotes. If you do not manually specify a path, the path is set to the default home directory that is the user_name home directory.
On IBM i operating systems:
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.

-jmport port_number
The JobManager port number used by the dynamic workload broker to
connect to the Tivoli Workload Scheduler dynamic agent. The default value is 31114. The valid range is from 1 to 65535.

-jmportssl true|false
The JobManager port used by the dynamic workload broker to connect to
the Tivoli Workload Scheduler dynamic agent. This number is registered in the ita.ini file located in the ITA\cpa\ita directory on Windows and in
the ITA/cpa/ita directory on UNIX, Linux, and IBM i.

For communication using SSL or HTTPS
Set jmportssl = 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 jmport communicates in HTTPS.

For communication without using SSL, or through HTTP
Set jmportssl = false. In this case the port specified in jmport
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.

Note: This is the language in which the installation log is recorded, and
not the language of the installed engine instance. The twsinst script installs all languages by 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 you entered in the thiscpu parameter. If not specified, the default value is MASTER.

-new A fresh installation of the agent. Installs an agent and all supported
language packs.

-password user_password
Windows only. The password of the user for which you are installing
Tivoli Workload Scheduler.

-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.

-reset_perm
UNIX and IBM i only. Reset the permission of the libraries in the
/usr/Tivoli directory.

-skip_usercheck
Enable this option if the authentication process within your organization is
not standard, thereby disabling the default authentication option. On
UNIX, skip the check of the user in the /etc/password file or using the su
command. On Windows, 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.

-stoponcheckprereq
Stop the installation whenever a problem occurs during the prerequisite
check. See "Checking prerequisites (UNIX and Linux)" on page 24 for
more information on the prerequisite check.

-tdwbservice host_name
The fully qualified host name of the dynamic workload broker. It is used
together with the -agent dynamic|both and the -tdwbservice tdwbservice_number
parameters. It is necessary to install the dynamic agent. If not specified,
you cannot run your workload dynamically and this parameter uses the
localhost default value. This value is registered in the ResourceAdvisorUrl
property in the JobManager.ini file.

-tdwbservice tdwbservice_number
The dynamic workload broker HTTP or HTTPS transport port number. It is
used together with the -agent dynamic|both and the -tdwbservice host_name
parameters. It is necessary to install the dynamic agent. This
number is registered in the ResourceAdvisorUrl property in the
JobManager.ini file. The default value is 31116. The valid range is from 0
to 65535. If you specify 0 or do not specify this parameter, you cannot run
workload dynamically. Do not specify 0 if the -agent value is dynamic or
both.

-thiscpu workstation
The name of the Tivoli Workload Scheduler workstation of this installation.
The name cannot exceed 16 characters, 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.

-u Displays command usage information and exits.

-uname username
The name of the user for which Tivoli Workload Scheduler is installed.
This user name is not the same as the user performing the installation
logged on as root on UNIX and Linux, as administrator on Windows, and
as QSECOFR on IBM i.

On UNIX and Linux, for a new installation, this user account must be
created manually before running the installation. Create a user with a
home directory. Tivoli Workload Scheduler is installed by default under the
home directory of the specified user.

On IBM i, 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.

Example installations
The following example shows the syntax used when using the twsinst script to
install a new instance of the fault-tolerant agent.

On UNIX and Linux:
On Windows operating systems:

```
./twsinst -new
-uname TWS_user
-adjruntime true
-agent both
-company IBM
-create_link
-hostname thishostname.mycompany.com
-inst_dir "c:\Program Files\IBM\TWA"
-jmport 31114
-master TWSmdm
-port 37124
-reset_perm
-stoponcheckprereq
-thiscpu mainworkstation
```

On Unix and Linux operating systems:

```
twinst -new
-uname TWS_user
-password user_password
-agent fta
-company IBM
-displayname thishostname.mycompany.com
-hostname thishostname.mycompany.com
-inst_dir "c:\Program Files\IBM\TWA"
-jmport 31114
-master TWSmdm
-port 37124
-thiscpu mainworkstation
```

The following example shows the syntax used when using the `twinst` script to install a new instance of the dynamic agent and adding the Java runtime for running job types with advanced options.

On Unix and Linux operating systems:

```
./twsinst -new
-uname TWS_user
-adjruntime true
-agent dynamic
-company IBM
-create_link
-hostname thishostname.mycompany.com
-inst_dir "c:\Program Files\IBM\TWA"
-jmport 31114
-master TWSmdm
-port 37124
-reset_perm
-stoponcheckprereq
-tdwport 31116
-tdwhostname mainbroker.mycompany.com
-thiscpu mainworkstation
```

On Windows operating systems:

```
twinst -new
-uname TWS_user
-password user_password
-adjruntime true
-agent dynamic
-company IBM
-displayname thishostname.mycompany.com
-hostname thishostname.mycompany.com
-inst_dir "c:\Program Files\IBM\TWA"
-jmport 31114
-master TWSmdm
```

Part 2. Tivoli Workload Scheduler
The following example shows the syntax used when using the `twsinst` script to install a new instance of both the fault-tolerant and the dynamic agent, and adding the Java runtime for running job types with advanced options.

On UNIX and Linux:
```
./twsinst -new
-uname TWS_user
-addjruntime true
-agent both
-company IBM
-create_link
-hostname thishostname.mycompany.com
-inst_dir "c:\Program Files\IBM\TWA"
-jmport 31114
-master TWSmdm
-port 37124
-reset_perm
-stoponcheckprereq
-tdwbbport 31116
-tdwbbhostname mainbroker.mycompany.com
-thiscpu mainworkstation
```

On Windows operating systems:
```
-twsinst -new
-uname TWS_user
-password user_password
-addjruntime true
-agent both
-company IBM
-displayname thishostname.mycompany.com
-hostname thishostname.mycompany.com
-inst_dir "c:\Program Files\IBM\TWA"
-jmport 31114
-master TWSmdm
-port 37124
-reset_perm
-stoponcheckprereq
-tdwbbport 31116
-tdwbbhostname mainbroker.mycompany.com
-thiscpu mainworkstation
```

**Installing agents using Software Distribution**

This section describes how to install Tivoli Workload Scheduler agents using Software Distribution software package blocks. During the installation, you can add the following:

- Standard agent, fault-tolerant agent, or domain manager capability
- Dynamic agent
- The Java runtime for running job types with advanced options, both the job types supplied with the product and the additional types implemented through the custom plug-ins.

The agent installed using the Software Distribution software package blocks has the following characteristics:

- It is installed in its own path, independent of any other Tivoli Workload Automation products or components installed on the same system.
- It cannot share components of the Tivoli Workload Automation network.
It cannot have a connector added to it and therefore cannot be directly connected to the Dynamic Workload Console.

Use Software Distribution software package blocks to install Tivoli Workload Scheduler agents only if you do not run a Java Virtual Machine on the workstation. If this is not your case, you might choose to perform a silent installation instead. See "Performing a silent installation" on page 84.

**Note:** Agents installed using Software Distribution software package blocks can only be uninstalled using Software Distribution.

**Software packages and parameters**
Tivoli Workload Scheduler agents can be installed distributing a software package block (SPB), using the Software Distribution component of Tivoli Configuration Manager, Versions 4.1, 4.2, 4.2.1, 4.2.2, or 4.2.3. You can distribute the SPB, using either the command line interface or from the Tivoli desktop.

**Note:** Do not modify the SPB supplied with the product.

An SPB exists for each supported operating system located on the installation disks under the directory of the operating system. The SPBs are named according to the operating system: Tivoli_TWS_<operating_system>.SPB. For the packages to be distributed, they must be imported in software package profiles. The software package profiles must be named according to the operating system and user: FP_TWS_<operating_system>_<TWS_user>.8.6.00. Possible values for operating system are:
- AIX
- HP
- SOLARIS
- WINDOWS
- LINUX_I386
- LINUX_PPC
- LINUX_S390
- SOLARIS_I386
- HPIA64
- LINUX_X86_64
- WINDOWS_X86_64

An SPB also exists to install language packs: Tivoli_TWS_LP.SPB. The software package profiles must be named according to the user: Tivoli_TWS_LP_<TWS_user>.8.6.00. The language pack software package block is located in the root directory of the installation DVD.

Tivoli Workload Scheduler installation parameters are defined as default variables in the software package. The following is a list of installation parameters.

**backup**
Optional. Indicates a backup. For a fresh installation, specify false. The default value is false.

**company**
Optional. The company name. This name appears in program headers and reports. The default value is COMPANY.

**display_name**
Required. The name to assign to the dynamic agent. The default is the host name of this computer.
**domain** Optional unless the user is a domain user. Windows operating systems only. The domain name of the user. The default value is `<computer_name>`.

**fresh_install**
Required. Indicates if this is a first time install. To perform a fresh installation, specify `false`. To perform an upgrade, specify `true`. The default value is `true`.

**from_release**
Required if you specified `upgrade`. When you specify `upgrade="true"`, you must also specify `from_release` indicating the release of the existing instance. The format is 8.x.

**group** Optional. The name of the group of operating system users.

**host_name** Optional. The fully qualified host name or IP address on which the agent is contacted by the Tivoli Dynamic Workload Broker.

**installer**
Optional. Windows operating systems only. The user ID of the installer of Tivoli Workload Scheduler. The default value is Administrator.

**install_dir**
Required. The fully qualified path to the directory of the Tivoli Workload Scheduler installation. This path must be a fully qualified path and cannot contain blanks. On Windows workstations, the path is created if it does not already exist. On UNIX and Linux operating systems, the path is the same as the user’s home directory. The default values are:
- Windows: `${system_drive}\win32app\TWS\<TWS_user>`
- UNIX and Linux: `user_home`

**jm_port**
Specify the value of the HTTP port that is used for communication between the Tivoli Workload Scheduler server and the Tivoli Workload Scheduler dynamic agent.

To communicate in HTTP you must also set the following parameters: `-D jm_port=nnnn` where `nnnn` is the port number and `-D jm_sec_port=0`. The default number of `jm_port` is 31114.

**jm_sec_port**
Specify the value of the HTTPS port that is used for communication between the Tivoli Workload Scheduler server and the Tivoli Workload Scheduler dynamic agent.

To communicate in HTTPS, you must also set the following parameters: `-D jm_port=0 -D jm_sec_port=nnnn` where `nnnn` is the port number. The default number of `jm_sec_port` is 31114.

**master_cpu**
Optional. The name of the master domain manager. The name cannot exceed 16 characters and cannot contain spaces. The default is `MASTER`.

**pwd** *(for Windows only)*
Required for Windows operating systems when installing for the first time. The password associated with the `<TWS_user>` user name. The SPB `password` variable is passed to the `pwd` variable.

**tcp_port**
Required. The Netman port used to run distributed scheduling. Netman is the network process that controls the production environment. When
installing more than one instance on the same workstation, use different port numbers for each instance. It must be an unassigned 16-bit value in the range from 1 to 65535. The default value is 31111.

**tdwb_hostname**
Optional. The dynamic workload broker fully qualified host name. Used together with the `-tdwbport <tdwbport_number>` parameter. Adds to Tivoli Workload Scheduler the capability to run workload dynamically. If not specified, you cannot run your workload dynamically and this parameter assumes the `localhost` default value. This value is registered in the `ResourceAdvisorUrl` property in the `JobManager.ini` file.

**tdwb_port**
Optional. The dynamic workload broker HTTP or HTTPS port number. Used together with the `-tdwbhostname host_name` parameter. Adds to Tivoli Workload Scheduler the capability to run workload dynamically. This number is registered in the `ResourceAdvisorUrl` property in the `JobManager.ini` file. The default value is 31116. The valid range is from 1 to 65535.

**this_cpu**
Required. The name of the workstation on which you are performing the installation. The name cannot exceed 16 characters and cannot contain spaces. The default value is THIS_CPU.

**tws_user**
Required. The user name for which Tivoli Workload Scheduler instance is being installed. On Windows systems, if this user account does not already exist, it is automatically created. If you specify a domain user or domain controller, you must specify the domain in the `domain` variable. If you specify a local user with the same name as a domain user, the local user must first be created manually by an administrator and then identified as `<system_name>\<user_name>`.

On UNIX and Linux operating systems, this user account must be created manually before running the installation.

**upgrade**
Required. Indicates if the installation is an upgrade. To perform an upgrade, specify `true`. To perform a fresh installation, specify `false`. The default value is `false`.

**Note:**
1. `fresh_install` and `upgrade` are mutually exclusive.
2. The variables that are not documented here are for debugging purposes only. See Administration Guide.

**Installation procedure**
The installation procedure checks that there is sufficient space for the Tivoli Workload Scheduler engine to be installed. It does not, however, check that there is sufficient space for the Configuration Manager backup directory specified in the `swdis.ini` file. See the Tivoli Configuration Manager documentation for these space requirements.

To perform the installation, complete the following steps:
1. Create a software package profile:
   FP_TWS_<operating_system>_TWS_<TWS_user>.8.6.00 where operating_system is the
   operating system where you are installing and <TWS_user> is the user of the
   installation.

2. Import the software package blocks using the wimspso command. When you
   import the software package blocks, you must pass the name of the profile to
   wimspso so that the Configuration Manager endpoint catalogs the name
   correctly.

3. Install the software package blocks using the wdinstsp command.

   Note: The supplied software packages must be installed as COMMITTED. The
   packages cannot be installed as UNDOABLE because the UNDO action does
   not rollback the product registry entries.

4. Depending on the type of agent you are installing, perform the steps in
   “Configuring a fault-tolerant agent” on page 164 and “Configuring a dynamic
   agent” on page 165.

Note: For complete instructions on performing these tasks, see wimspso and
   wdinstsp in the IBM Tivoli Configuration Manager, Reference Manual for
   Software Distribution, and the IBM Tivoli Configuration Manager, User’s Guide
   for Software Distribution.

Prerequisite: Installing the Common Inventory Technology (CIT):
You must install CIT before installing the agent and adding the following
   capabilities:
   • Standard agent, fault-tolerant agent, or domain manager capability
   • Dynamic scheduling capability
   • The option to add the runtime for Java environment jobs

The following are examples of the commands you run to install CIT on Windows
   and UNIX workstations. See “Software packages and parameters” on page 97
   for a description of the parameters.

Windows operating systems:
   1. wdinstsp -D CIT_ExploiterID=TWA D:\TWS_86\WINDOWS\CIT_Preinstall.spb
   2. wdinstsp D:\TWS_86\WINDOWS\CIT.spb

UNIX and Linux operating systems:
   1. wdinstsp -D CIT_ExploiterID=TWA /TWS_86/UNIX/CIT_Preinstall.spb
   2. wdinstsp /TWS_86/UNIX/CIT.spb

Installing the Tivoli Workload Scheduler dynamic agent:
This section describes how to install the Tivoli Workload Scheduler dynamic agent
   by using the wdinstsp command. In addition, you can add the Java runtime to
   the agent to run job types with advanced options, both those types supplied with
   the product and the additional types implemented through the custom plug-ins See
   “Adding the Java runtime” on page 102.

To install the Tivoli Workload Scheduler dynamic agent, perform the following
   steps:
   1. Verify the authorizations required to run the procedure in the “User
      authorization requirements” on page 59 section.
   2. Locate the .spb as described in the “Software packages and parameters” on
      page 97 section.
3. Set the Software Distribution environment launching the following command located in the TWA/TWS/_uninstall/CLI directory:

**Windows operating systems:**

swd_env.bat

**UNIX and Linux operating systems:**

swd_env.sh

4. Run the `wdinstsp` command located in the TWA/TWS/_uninstall/CLI as shown in the following examples:

**Windows operating systems**

The following Windows example describes an installation with the user `<TWS_user>` and the endpoint *Tivoli_TWS_WINDOWS*. In this example, you are installing on a domain controller or in a Windows node agent because the `-D domain="domain_name"` was specified.

```plaintext
wdinstsp
    -f
    -uy
    -D install_dir="C:\ibm\TWS\twuser\TWS"
    -D tws_user="twuser"
    -D password="twspasswd"
    -D domain="domain_name"
    -D group="group_name"
    -D installer="Administrator"
    -D jm_port="0"
    -D jm_sec_port="31114"
    -D host_name="IT041924-T61.rot.ibm.com"
    -D display_name="IT041924-T61_1"
    -D tdwb_port="31116"
    -D "tdwb_hostname=slt.romelab.it.ibm.com"
    -D fresh_install="true"
    -D upgrade="false"
    -D execActionTools="true"
    -D startAgent="true"
    -n "TWS_LWA_twuser.8.6.0.00"
    "C:\Output\TWS_VLAST\WINDOWS\Tivoli_LWA_WINDOWS.SPB"
```

**UNIX and Linux operating systems**

The following UNIX example describes an installation with the user `<TWS_user>` and the endpoint *Tivoli_TWS_LINUX_I386*.

```plaintext
wdinstsp
    -f
    -uy
    -D install_dir="/home/twuser/TWS"
    -D tws_user="twuser"
    -D domain="null"
    -D group="group_name"
    -D installer="root"
    -D jm_port="0"
    -D jm_sec_port="31114"
    -D host_name="IT041924-T61.rot.ibm.com"
    -D display_name="IT041924-T61_1"
    -D tdwb_port="31116"
    -D "tdwb_hostname=slt.romelab.it.ibm.com"
    -D fresh_install="true"
    -D upgrade="false"
    -D execActionTools="true"
    -D startAgent="true"
    -n "TWS_LWA_twuser.8.6.0.00"
    /mnt/gsa/home/s1/user1/web/public/SPB_INSTALL/
    LINUX_I386/Tivoli_LWA_LINUX_I386.SPB"
```
The following are examples of the settings required to perform a fresh installation of an agent on Windows and UNIX workstations. See “Software packages and parameters” on page 97 for a description of the parameters.

Adding the Java runtime:

Add the Java runtime to add the following functions to your agent:

- 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 remotely run, from the agent, the dynamic workload broker resource command on the server.

To add the Java runtime, perform the following steps:

1. Verify the authorizations required to run the procedure in the “User authorization requirements” on page 59 section.
2. Locate the .spb as described in the “Software packages and parameters” on page 97 section.
3. Set the Software Distribution environment launching the following command located in the TWA/TWS/_uninstall/CLI directory:
   
   Windows operating systems:
   
   swd_env.bat

   UNIX and Linux operating systems:
   
   swd_env.sh

4. Run the wdinstsp command located in the TWA/TWS/_uninstall/CLI as shown in the following examples:

   See “Software packages and parameters” on page 97 for a description of the parameters.

   Windows operating systems:
   
   wdinstsp
   
   -f
   -uy
   -D install_dir="C:\ibm\TWS\twuser\TWS"
   -D tws_user="twuser"
   -D group="group_name"
   -D installer="Administrator"
   -n "TWS_Eclipse_twuser.8.6.0.00"
   "C:\Output\TWS_VLAST\WINDOWS\Tivoli_Eclipse_WINDOWS.SPB"

   UNIX and Linux operating systems:
   The following UNIX example describes an installation with the user <TWS_user>:

   wdinstsp
   
   -f
   -uy
   -D install_dir="/home/twuser/TWS"
   -D tws_user="twuser"
   -D group="group_name"
   -D installer="root"
   -n "TWS_Eclipse_twuser.8.6.0.00"
   
   /mnt/sga/home/s1/user1/web/public/SPB_INSTALL/
   LINUX_I386/Tivoli_Eclipse_LINUX_I386.SPB

Adding standard agent, fault-tolerant agent, or domain manager capability:

To add standard agent, fault-tolerant agent, or domain manager capability to the Tivoli Workload Scheduler agent on Windows and UNIX workstations, perform the following steps.
1. Verify the authorizations required to run the procedure in the “User authorization requirements” on page 59 section.

2. Locate the .spb as described in the “Software packages and parameters” on page 97 section.

3. Set the Software Distribution environment launching the following command located in the TWA/TWS/_uninstall/CLI directory:

   Windows operating systems:
   
   swd_env.bat

   UNIX and Linux operating systems:
   
   swd_env.sh

4. Run the wdinstsp command located in the TWA/TWS/_uninstall/CLI as shown in the following examples:

   See “Software packages and parameters” on page 97 for a description of the parameters.

   Windows operating systems:
   
   In the following example, you are installing on a domain controller or in a Windows node agent because the -D domain=”domain_name” parameter was specified. The installation is performed with the user <TWS_user> and the endpoint Tivoli_TWS_WINDOWS.

   wdinstsp
   
   -n "FP_TWS_WINDOWS_twsuser.8.6.0.00"
   -D install_dir="C:\IBM\TWS\twsuser\TWS"
   -D tws_user="twsuser"
   -D password="twspasswd"
   -f
   -uy
   -D company="company_name"
   -D this_cpu="IT041924-T61"
   -D master_cpu="MTMDM"
   -D tcp_port="33311" jm_port=0
   -D jm_sec_port=31114
   -D domain="domain name"
   "C:\Output\TWS_VLAST\WINDOWS\Tivoli_TWS_WINDOWS.SPB"

   UNIX and Linux operating systems
   
   The following UNIX example describes an installation with the user <TWS_user> and the endpoint Tivoli_TWS_LINUX_I386.

   wdinstsp
   
   -n FP_TWS_WINDOWS_twsuser.8.6.0.00
   -f
   -uy
   -D install_dir="/home/twsuser/TWS"
   -D tws_user="twsuser"
   -D company="company_name"
   -D this_cpu="IT041924-T61"
   -D master_cpu="MTMDM" jm_port=0
   -D jm_sec_port=31114
   -D tcp_port="33311"
   -D serverName="server1"
   /mnt/gsa/home/s1/user1/web/public/SPB_INSTALL/
   LINUX_I386/Tivoli_TWS_LINUX_I386.SPB

Installing language packs

You can install language packs using Software Distribution. Locate the Tivoli_TWS_LP.SPB software package block in the root directory of the DVD, and then customize the following parameters before you install.
Table 7. List of parameters to install language packs

<table>
<thead>
<tr>
<th>Default variable</th>
<th>Description</th>
<th>Required</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>zh_CN</td>
<td>Chinese, Simplified</td>
<td></td>
<td>false</td>
</tr>
<tr>
<td>it</td>
<td>Italian</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ko</td>
<td>Korean</td>
<td></td>
<td></td>
</tr>
<tr>
<td>es</td>
<td>Spanish</td>
<td></td>
<td></td>
</tr>
<tr>
<td>zh_TW</td>
<td>Chinese, Traditional</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ja</td>
<td>Japanese</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pt_BR</td>
<td>Brazilian Portuguese</td>
<td></td>
<td></td>
</tr>
<tr>
<td>de</td>
<td>German</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fr</td>
<td>French</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALL_LANG</td>
<td>All of the above languages.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tws_user</td>
<td>The user name for which the specified language pack is being installed.</td>
<td>Yes</td>
<td>$(user_name)</td>
</tr>
<tr>
<td>install_dir</td>
<td>The fully qualified path to which the specified language packs are installed.</td>
<td>Yes</td>
<td>$(program_files)</td>
</tr>
</tbody>
</table>

1. Verify the authorizations required to run the procedure in the "User authorization requirements" on page 59 section.
2. Locate the .spb as described in the "Software packages and parameters" on page 97 section.
3. Set the Software Distribution environment launching the following command located in the TWA/TWS/_uninstall/CLI directory:

   **Windows operating systems:**
   ```
   swd_env.bat
   ```

   **UNIX and Linux operating systems:**
   ```
   swd_env.sh
   ```

4. Run the `wdinstsp` command located in the TWA/TWS/_uninstall/CLI as shown in the following examples: The following is the syntax required to install all languages:

   ```
   wdinstsp
   -D install_dir="Installation Path"
   -D tws_user="Username"
   [-D zh_C =true ...
    -D de=true | ALL_LANG=true]
   Tivoli_TWS_LP.SPB [subscribers...]
   ```

   The following is the syntax required to install Italian and German language packs:

   ```
   wdinstsp
   -D install_dir="Installation Path"
   -D tws_user="Username"
   [-D it =true | -D de=true]
   Tivoli_TWS_LP.SPB [subscribers...]
   ```
Installing the Job Brokering Definition Console

This section describes how to install the Job Brokering Definition Console. It is divided into the following topics:

- "Installing the Job Brokering Definition Console using the installation wizard"
- "Performing a silent installation of the Job Brokering Definition Console"

The Job Brokering Definition Console is a structured editing tool that you use to create and modify Job Submission Description Language (JSDL) files. These files are saved in the Job Repository as job definitions and become available for submission. The JSDL files adhere to the XML syntax and semantics as defined in the JSDL schema. For more information about the Job Brokering Definition Console, see the *Tivoli Workload Scheduler: User’s Guide and Reference, SC32-1274*.

The Job Brokering Definition Console is supported only on Windows 32-bit and Linux 32-bit. You can install one instance of the Job Brokering Definition Console for a single user on each workstation. This is because two instances installed by the same user share the same workspace. If you need to install two instances of the Job Brokering Definition Console on the same workstation, install each instance using a different user and ensure that each instance accesses its own workspace.

**Installing the Job Brokering Definition Console using the installation wizard**

For a graphical installation, from the installation DVD, start the launchpad as described in "Launchpad" on page 30 and select the Job Brokering Definition Console installation, or run the setup for the operating system on which you are installing.

From the root directory of the DVD, run the following:

- On Windows operating systems: JBDC\WORKBENCH\setupwin32.exe
- On Linux operating systems: JBDC/WORKBENCH/setuplinux.bin

Follow the installation wizard, providing the installation directory name, to complete the installation.

**Performing a silent installation of the Job Brokering Definition Console**

For a silent installation, copy the following file to a local directory:

<images_path>/JBDC/WORKBENCH/ResponseFiles/TDWB_Workbench_installation.rsp

In this file, edit the following parameters:

- `licenseAccepted=true`
- `installLocation="<installation_path>"`

To perform a silent installation using a response file template, enter the following command:

`-options "<path-to-ResponseFile>/TDWB_Workbench_installation.rsp" -silent`

For information about response files and silent installation, see "Performing a silent installation" on page 84.
Installing the additional plug-ins by using the Tivoli Workload Scheduler for Additional Plug-ins

This section describes how to install one or more additional plug-ins by using the Tivoli Workload Scheduler for Additional Plug-ins. It is divided into the following topics:

- “Before installing” on page 107
- “Selecting your installation method” on page 107
- “Installing by using the installation wizard” on page 107
- “Installing by using the silent installation” on page 108

The Tivoli Workload Scheduler for Additional Plug-ins is an installation process that you can use to install the additional plug-ins that you have developed to resolve your particular needs. This installer is contained in the Tivoli Workload Scheduler Fix Pack 1 DVD or eImages.

Before installing

Before you install the additional plug-in, ensure that the following conditions are satisfied:

- You must have the following structure for the plug-in file:
  `<plug-in_namespace>.<plug-in_id>_<plug-in_version>.zip`
  as described in the following section: "The additional plug-in structure."

- You must have the following permissions to run the installation:
  - Windows operating systems:
    Administrator
  - UNIX and Linux operating systems:
    root
  - The installation process is not already running on the workstation. You can verify it by checking that the `setup` process is not running.

The additional plug-in structure

This section describes the plug-in structure.

- The plug-in zip name must be the following:
  `<plug-in_namespace>.<plug-in_id>_<plug-in_version>.zip`

- You must have the following structure for the plug-in file:
  `<plug-in_namespace>.<plug-in_id>_<plug-in_version>.zip`:

  /files/license
  /files/plugin.xml
  /files/<plug-in_name>.properties
  /files/<plug-in_name>_<plugin_version>.jar

  where,
  - The /files/license directory contains the License agreement files. This directory is optional.
  - The optional file `<plug-in_name>.properties` contains the properties of the plug-in.
  - The mandatory file /files/plugin.xml must have the following structure:

    <plugin>
      <pluginInfo version="<plugin_version>" name="<plug-in_name>"
        id="<plug-in_ID>" />
      <vendor name="<company_name>" id="<company_id>" />
      <pluginInstaller minVersionSupported="<plugin_min_version>" />
      <twsInstances>
        <twsInstance version="8.6.0.00" />
    </plugin>
Selecting your installation method

You can install the additional plug-ins by using one of the methods described in this section. To install a additional plug-in, use any of the following procedures. If you want to install another one, start the installation procedure again.

Installation wizard

Install additional plug-in on an existing installation by running the individual setup files for each supported operating system. For details, see “Installing by using the installation wizard.”

Silent installation

Customize a response file by adding all the configuration settings to be used during installation. Then, from the command line, run the setup command. With this procedure, you can run the installation unattended and in the background. For details, see “Installing by using the silent installation” on page 108.

Before starting the installation process, ensure that the file <plug-in_namespace>.<plug-in id>_<plug-in_version>.zip is built as described in “Before installing” on page 106.

Note: To successfully use the installed plug-ins, you must first restart WebSphere Application Server, Tivoli Workload Scheduler agent or both.

Installing by using the installation wizard

To install additional plug-in, perform the following steps:

1. From the Tivoli Workload Scheduler Fix Pack 1 DVD or eImages, for the operating system where you are installing, run the setup installation command. It is located in the /PLUGIN_INSTALLER directory. The installation starts.

2. Select the language in which you want the wizard to be displayed, and click OK. The Welcome panel is displayed.

3. Read the welcome information and click Next. The operations panel is displayed.

4. Select the Install radio button. Click Next. The zip file location panel is displayed.

5. Select the path on your workstation where the zip file <plug-in_namespace>.<plug-in id>_<plug-in_version>.zip is located. If the installation program does not detect any zip file in the path you specified, you cannot perform any actions.

6. Click Next. The plug-in details panel is displayed.
7. Review the plug-in details, and click **Next**. The plug-in Software License Agreement panel is displayed.

8. Read the plug-in Software License Agreement information and select the radio button to accept the license agreement. Click **Next**. A summary information panel is displayed.

9. Select the Tivoli Workload Scheduler on your workstation where the zip file `<plug-in_namespace>.<plug-in_id>_<plug-in_version>.zip` is installed.
   If the installation program does not detect any instance of Tivoli Workload Scheduler, you cannot perform any actions.

10. Review the summary details and click **Install**. The installation process begins; the progress panel is displayed showing the status.

If you received error messages, analyze the installation log files shown in the table [Table 9 on page 111](#).

**Installing by using the silent installation**

A silent installation runs according to the parameters set in a response file. The response file includes all the installation information required to run the installation without user intervention.

To install additional plug-in with the silent installation, you are provided with the following response file located under `PLUGIN_INSTALLER/RESPONSE_FILE` on the product DVD:

`TWS_Plug-ins_RespFile_<operatingsystem>.txt`

where `<operatingsystem>` can be unix or windows.

It is a template file that you can customize to reflect the additional plug-in you want to install.

**Note:** Using the silent installation you can install one plug-in at time.

When running the installation in silent mode, no messages are displayed. The messages are written in the silent installation log files listed in [“Installation actions and log files” on page 110](#). If the silent installation fails, you can verify the messages written in the log files, by checking them in the section [“Analyzing return codes for Tivoli Workload Scheduler for Additional Plug-ins silent installation” on page 191](#).

To run the silent installation, perform the following steps:

1. Create your response file or customize the response file to include the options required to complete the installation.

   `TWS_Plug-ins_RespFile_<operatingsystem>.txt`

   For a list of these options, see [Table 8 on page 109](#).

   The response file must be accessible from the workstation where you want to perform the installation. Entries in the response file are in the format `option=value`. Each entry must be written on a separate line.

2. Insert the product DVD for your operating system and run the **setup** command, located in the `PLUGIN_INSTALLER/` directory:

   **On UNIX and Linux operating systems:**
   
   ```bash
   ./setup.sh -i silent -f response_file
   ```

   **On Windows operating systems:**

   ```cmd
   setup silent /f response_file
   ```
setup.bat -i silent -f response_file

Where:

-i silent
Specifies that the installation is run unattended, driven by a response file.

-f response_file
Indicates the fully qualified path to the response file that contains the installation options. response_file can be any text file with the name and extension you choose.

The actions performed by installation is described in the section “Installation actions and log files” on page 110.

Table 8 lists the options you can specify to drive the installation.

<table>
<thead>
<tr>
<th>Option</th>
<th>Required</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
</table>
| USER_INSTALL_DIR=<path>     | Yes      | Specify the Tivoli Workload Scheduler installation path where you want to install an additional plug-in. | A fully qualified path. For example, to install the product under c:\program Files\IBM\TWA86, specify: USER_INSTALL_DIR="c:\program Files\IBM\TWA86"

The product files are installed in:

c:\program Files\IBM\TWA86\methods

On Windows operating systems:
The default path is "c:\\Program Files\IBM\TWA"

On UNIX and Linux operating systems:
The default path is /opt/IBM/TWA

TWSAPPS_PLUGIN_FILE_NAME=<zip-filename> | Yes      | Specify the fully qualified path to the zip file that contains the addition plug-in that you want to install. | A fully qualified path. For example, to install the additional plug-in <test_plug-in>.zip located in C:\Documents and Settings\Administrator\Desktop\PLUGINS, specify: TWSAPPS_PLUGIN_FILE_NAME=C:\Documents and Settings\Administrator\Desktop\PLUGINS\<test_plug-in>.zip

LICENSE_ACCEPTED=<value> | Yes      | Specify the boolean value to accept license agreement of additional plug-in. | The value can be true or false. But the plug-in installation proceed even if the value is set to true

Part 2. Tivoli Workload Scheduler 109
Table 8. Options to perform a silent installation (continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Required</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTION_TYPE=&lt;value&gt;</td>
<td>Yes</td>
<td>Specify the action that installation process performs on plug-in. In this case the value must be set to DEPLOY.</td>
<td>The value must be set to DEPLOY.</td>
</tr>
</tbody>
</table>

The following is an example of the command you run to perform a silent installation on a UNIX workstation, by using the response file TWSPlug-ins_RespFile_unix.txt:

```
./setup.sh -i silent -f /tmp/TWSPlug-ins_RespFile_unix.txt
```

The following example shows a response file that installs the additional plug-in contained in the zip file <plug-in>.zip on a Windows workstation:

```
USER_INSTALL_DIR="c:\Program Files\IBM\TWA"
TWSAPPS_PLUGIN_FILE_NAME=C:\Documents and Settings\Administrator\Desktop\PLUGINS\<plug-in>.zip
PLUGINS\<plug-in>.zip
```

**Installation actions and log files**

This section describes the additional plug-in installation process actions and installation logs files.

The additional plug-in installation is possible on Tivoli Workload Scheduler instance of:

- Master domain manager
- Backup master domain manager
- Dynamic Domain manager
- Backup Dynamic Domain manager
- Agent
- Fault-tolerant agent with Java extension installed.

The structure of the zip file is described in the section "The additional plug-in structure" on page 106.

The installation performs the following actions on the content of the zip file <plug-in_namespace>.<plug-in id>_<plug-in_version>.zip for all Tivoli Workload Scheduler workstation types:

- Copy the file plugin.xml as plugin_<plugin_name>.xml in the directory <TWA_HOME>/installDataPlugins of the selected Tivoli Workload Scheduler instance.
- Copy the file /files/<plug-in_name>.properties in the directory <TWA_HOME>/TWS/JavaExt/eclipse/configuration of the selected Tivoli Workload Scheduler instance.
- Copy the file /files/<plugin_namespace>.<plugin_id>_<plugin_version>.jar in the directory <TWA_HOME>/TWS/JavaExt/eclipse/configuration of the selected Tivoli Workload Scheduler instance.
- Copy all the files in the /files/licenses directory in the directory <TWA_HOME>/license/<plug-in_id> of the selected Tivoli Workload Scheduler instance.
The installation also performs the following actions on the content of the zip file `<plug-in_namespace>.<plug-in id>_<plug-in_version>.zip` for specific workstation types:

- For **master domain manager**, **backup master domain manager**, **Dynamic Domain manager**, and **Backup Dynamic Domain manager**, also copy the file `files/<plugin_namespace>.<plug-in_id>_<plugin_version>.jar` in the directory `<TWA_HOME>/TWS/applicationJobPlugins` of the selected Tivoli Workload Scheduler instance.

- For **Tivoli Workload Scheduler for z/OS connector**, also copy the file `files/<plugin_namespace>.<plug-in_id>_<plugin_version>.jar` in the directory `<TWA_HOME>/TWSZOS/applicationJobPlugins` of the selected Tivoli Workload Scheduler instance.

If you received error messages, analyze the installation log files shown in Table 9.

<table>
<thead>
<tr>
<th>Log file name</th>
<th>Content</th>
<th>Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>tws4plugins_ia_install.log</td>
<td>additional plug-in log file for InstallAnywhere errors.</td>
<td>At the begin of the installation process this log file is created in the following directory:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>On Windows operating systems:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>%TEMP%	wa	ws4apps</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>On UNIX and Linux operating systems:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$tmp	wa	ws4apps and copied to directory Tivoli Workload Scheduler_installation_dir\logs at the end of the installation process.</td>
</tr>
<tr>
<td>tws4plugins_install.log</td>
<td>The additional plug-in installation log file.</td>
<td>At the begin of the installation process this log file is created in the following directory:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>On Windows operating systems:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>%TEMP%	wa	ws4apps</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>On UNIX and Linux operating systems:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$tmp	wa	ws4apps and copied to directory Tivoli Workload Scheduler_installation_dir\logs at the end of the installation process.</td>
</tr>
<tr>
<td>tws4plugins_status.log</td>
<td>The additional plug-in installation status log file is created only for silent installation. It reports if the installation completed successfully or with errors. In case of errors it indicates if the error is due to an incorrect field value or to a failed step.</td>
<td>At the begin of the installation process this log file is created in the following directory:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>On Windows operating systems:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>%TEMP%	wa	ws4apps</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>On UNIX and Linux operating systems:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$tmp	wa	ws4apps and copied to directory Tivoli Workload Scheduler_installation_dir\logs at the end of the installation process.</td>
</tr>
</tbody>
</table>

**Note:** If you are installing in silent mode and you need to see the logs files, check before the `tws4plugins_status.log` file to verify the installation process status and then check the `tws4plugins_install.log` file for details.
This chapter describes how to upgrade Tivoli Workload Scheduler from version 8.3 and higher to the current version. It is divided into the following sections:

- "Engine coexistence and upgrade notes"
- "User authorization requirements" on page 59
- "Checking prerequisites (UNIX and Linux)" on page 114
- "Upgrading a master domain manager or backup master domain manager instance" on page 114
- "Upgrading agents and domain managers" on page 141
- "Upgrading a command line client" on page 155
- "Upgrading when there are corrupt registry files" on page 156
- "Adding a feature" on page 158

Note: Only when the database administrator manages all the confidential information related to the database, such as the database administrator user ID and password and the IT administrator who upgrades the product does not know them, run the procedure described in "Creating or upgrading the Tivoli Workload Scheduler database tables before installing or upgrading" on page 36.

**Engine coexistence and upgrade notes**

This section contains information about coexistence with older versions and upgrade possibilities.

**Coexistence with previous versions**

The current version of the Tivoli Workload Scheduler distributed engine can be installed on any workstation containing a prior version, provided that both the `<TWS_user>`, the nm port, and the installation path are different from those of the previous versions.

**Upgrading existing versions**

The upgrade of a Tivoli Workload Scheduler network can be performed top-down or bottom-up. The advantages and disadvantages of these approaches are discussed in "Choosing how to migrate your network" on page 115.

Table 10 shows the versions of Tivoli Workload Scheduler components that can be upgraded to the current version.

<table>
<thead>
<tr>
<th>Component</th>
<th>Version</th>
<th>Minimum recommended fix pack level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master domain manager</td>
<td>8.3</td>
<td>7 and higher</td>
</tr>
<tr>
<td>Backup master domain manager</td>
<td>8.4</td>
<td>GA and higher</td>
</tr>
<tr>
<td>Fault-tolerant agent</td>
<td>8.5</td>
<td>GA and higher</td>
</tr>
<tr>
<td></td>
<td>8.5.1</td>
<td>GA and higher</td>
</tr>
<tr>
<td>Command-line client</td>
<td>8.5.1</td>
<td>GA and higher</td>
</tr>
<tr>
<td>Connector</td>
<td>Upgrade available from version 8.3</td>
<td></td>
</tr>
</tbody>
</table>

- If you are using Tivoli Workload Scheduler components at a lower fix pack level, install the fix pack at the indicated level, or higher.
If you are using Tivoli Workload Scheduler components from a previous version not supported in Table 10 on page 112, consider replacing them with a fresh installation of the latest components, or upgrading them as follows:

1. Upgrade them to one of the supported upgrade platforms, using the upgrade programs and procedures documented for that platform.
2. Apply the necessary fix packs, as shown in Table 10 on page 112.
3. Upgrade them to the current version.

**Files and folders changed during the upgrade**

The upgrade process changes the following folders and files:

**On UNIX and Linux operating systems:**

```
/etc/TWS
/etc/TWA
./swdis (This is the default Software Distribution directory. The product changes the directory specified in the product_dir property in the /etc/Tivoli/swdis.ini file.)
/usr/Tivoli/TWS
tws_home
```

**On Windows operating systems:**

```
%windir%\TWA
%windir%\system32\TWS*Registry.dat
C:\swdis (This is the default Software Distribution directory. The product changes the directory specified in the product_dir property in the %windir%\swdis.ini file.)
tws_home
```

**User authorization requirements**

Check the authorization roles before beginning the procedure.

Authorization requirements for running any of the installation, upgrade, or uninstallation wizards or commands are the same:

**UNIX and Linux operating systems**

- root access

**Windows operating system**

Your login account must be a member of the Windows Administrators group or domain administrators with 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. To do this, before running the installation, perform the following steps:

1. Right-click the icon that you use to run the program:
   - *If you are using the wizard:*
     - SETUP.exe
   - *If you are using the silent installation or twsinst:*
     - Command Prompt
2. Select Run as administrator.

In addition, to use Software Distribution, the user must have the Software Distribution roles admin, senior, or super.
Checking prerequisites (UNIX and Linux)

If you are preparing to upgrade on UNIX and Linux operating systems, Tivoli Workload Scheduler automatically runs a prerequisite check on your system. Having an environment that meets the Tivoli Workload Scheduler system requirements ensures that an upgrade succeeds without any delays or complications.

Note: The prerequisite check is not available if you are using the Software Distribution installation method and is only available on UNIX and Linux operating systems.

The prerequisite check verifies that:
- The operating system is supported for the product.
- The necessary engine software patch levels are installed.
- The necessary kernel parameters are correctly configured.

Note: The prerequisite check only verifies that the environment meets the requirements of the Tivoli Workload Scheduler. It does not check the installation requirements for other components, such as DB2.

For a detailed description on how the check works, see “Checking prerequisites (UNIX and Linux)” on page 24.

Upgrading a master domain manager or backup master domain manager instance

This section describes how to upgrade master domain managers and backup master domain managers.

This section is divided into the following topics:
- “Upgrading overview”
- “Preparing to upgrade” on page 122
- “New directory structure” on page 124
- “Performing a direct upgrade” on page 126
- “Performing a parallel upgrade” on page 137

Upgrading overview

This section provides an overview of the upgrade of an existing version of Tivoli Workload Scheduler V8.3 and higher instance. It is divided into the following sections:
- “Choosing how to migrate your network” on page 115
- “Component upgrade procedures” on page 115
- “Parallel or direct upgrading” on page 116
- “Updating authentication” on page 118
- “Performing a safe upgrade” on page 121

Note: When you upgrade from version 8.5, you must upgrade the entire instance of Tivoli Workload Automation. For information about Tivoli Workload Automation instances, see “Instances of Tivoli Workload Automation” on page 27.
To upgrade an instance of Tivoli Workload Automation, you must upgrade all components that are part of that instance, starting with the components of Tivoli Workload Scheduler first. For example, if your instance includes one master domain manager and also the Dynamic Workload Console, you must upgrade both of these components but you must upgrade the master domain manager first and then the Dynamic Workload Console next.

**Choosing how to migrate your network:**
Tivoli Workload Scheduler supports backwards compatibility so you can decide to upgrade your network in either of the following ways:

**Top-down**
Upgrade the master domain manager and backup master domain manager, and then progressively upgrade the agents. Many of the new functions 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. Even if you distribute the new security file to an agent prior to V8.5, it does not function until you upgrade it. You must upgrade any agents that are older than V8.5.

**Bottom-up**
Upgrade the agents first, and then upgrade the master domain manager and backup master domain manager. The new functions introduced in the current version are not available until the whole network has been upgraded.

**Component upgrade procedures:**
Figure 11 on page 116 is a flowchart of the steps you can perform to upgrade Tivoli Workload Scheduler to the current version. Note the different pathways you follow for a direct and a parallel upgrade. These options are discussed in detail following the chart.

The following acronyms are used in the flowchart:

**MDM** Master domain manager

**BKM** Backup master domain manager
Parallel or direct upgrading:

Figure 11. Upgrade procedure flowchart
There are several factors to consider before you upgrade your master domain manager and backup master domain manager. This section describes these factors and outlines the available upgrade scenarios from which you must choose:

- "Direct upgrade scenario - minimizing the time to upgrade"
- "Parallel upgrade scenario - minimizing the impact on scheduling"

On balance IBM believes that the direct upgrade scenario is the preferable option, especially if you are upgrading from V8.5 or V8.5.1, and assuming that you have a backup master domain manager available for use in case of unforeseen problems. However, both scenarios are fully described and supported and you must make your own choice considering the particular factors at work in your scheduling environment.

**Direct upgrade scenario - minimizing the time to upgrade:**
The direct upgrade scenario allows you to upgrade your current environment quickly, reducing manual intervention. The procedure automatically upgrades your network and database information using the input you provide. The installation wizard is the simplest way of approaching this type of upgrade because it guides you through the process.

**Table 11. Direct upgrade scenario**

<table>
<thead>
<tr>
<th>Steps</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Unlink the old master domain manager and then stop it.</td>
<td>• Quicker and simpler than the parallel upgrade.</td>
<td>• Scheduling might be delayed for those activities involving the master domain manager.</td>
</tr>
<tr>
<td>2. Upgrade the master domain manager, automatically importing the scheduling and configuration data from the previous version.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Complete the security configuration by merging old and new security settings.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Upgrade the backup master domain manager.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

However, you are strongly recommended to ensure that you have a backup master domain manager installed, tested, and available to takeover immediately, in the event that the upgrade fails and the master domain manager is not left in a usable state.

**Parallel upgrade scenario - minimizing the impact on scheduling:**
A parallel upgrade allows you to maintain the integrity of your previous master domain manager until you are confident with the new environment. The upgrade is staged and allows you to work in coexistence with your old environment.

In the parallel scenario described in the following sections, you start by upgrading your existing backup domain manager or by installing a new Tivoli Workload Scheduler backup domain manager. Your new or upgraded backup master domain manager then assumes the role of your old master domain manager. You then have the choice of making this new environment permanent. Alternatively, you can upgrade and restore the old master domain manager to its original role.

This sequence of operations is designed to minimize your out-of-service time and to ensure data integrity. The parallel upgrade involves a limited number of manual steps but has the advantage of maintaining the integrity of your current environment.
### Table 12. Parallel upgrade scenario

<table>
<thead>
<tr>
<th>Steps</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
</table>
| 1. Perform one of the following depending on whether or not you already have a backup master domain manager:  
  - Upgrade your current backup master domain manager referencing the existing database.  
  - Install a fresh backup master domain manager, that points to the existing database. You also need to migrate your user authentication configuration to the new backup master domain manager. |  
  - Allows the coexistence of the old master with the new environment.  
  - Allows you to choose a new and better performing platform for your new master domain manager.  
  - It is a reversible process.  
  - Automatically updates the database schema which is fully compatible for old and new versions. |  
  - Involves some manual configuration steps. |
| 2. Switch your manager to the new backup master domain manager.      |                                                                           |                                                                               |
| 3. Make the switch manager permanent.                                |                                                                           |                                                                               |
| 4. Your backup master domain manager has now become the master domain manager of the new environment. Before you proceed to the next step, decide what to do with your old master domain manager. You have three alternatives:  
  - Keep the new manager as the master domain manager of your new environment and your old master domain manager as a full status agent, upgrading it later to the new version. Install a new backup master domain manager and configure it.  
  - Keep the new manager as the master domain manager of your new environment and upgrade the old master to become the new backup master domain manager.  
  - Upgrade your old master domain manager and restore the original configuration in the new environment (this is the procedure indicated in Figure 11 on page 116) |  
  - Allows a great degree of flexibility. You can choose to not upgrade the old master domain manager.  
  - You can easily upgrade hardware at the same time you perform the parallel upgrade. |                                                                               |
| 5. If you switch back, and your original switch was made permanent, you must make the switch back permanent as well |                                                                           |                                                                               |
| 6. Complete your security configuration by merging old and new security settings. |                                                                           |                                                                               |

**Updating authentication:**

This section describes how your configured authentication mechanism is upgraded.

In versions of Tivoli Workload Scheduler before V8.6, authentication was configured to use stand-alone user registries, managed by the embedded WebSphere Application Server. The available options were:

- Local operating system
- Custom (through PAM - Pluggable Authentication Module)
If you enabled LDAP, you could use one of the following servers:

- IBM Tivoli Directory Server
- Sun ONE
- Microsoft Windows Active Directory
- RACF® configured on IBM Tivoli Directory Server

Versions of Tivoli Workload Scheduler from V8.6 onwards are configured for authentication (through the embedded WebSphere Application Server) in VMM (Virtual Member Manager) mode. This creates a **Federated User Registry**, which supports the simultaneous use of more than one user registry. The user registry choices and LDAP server options are similar to those in versions before V8.6.

The implementation of the Federated User Registry, however, requires a different configuration than the stand-alone configuration you are currently using.

During the upgrade, your existing configuration is migrated, so that when the upgrade is complete the product is configured to use the same authentication mechanism as before, but within a Federated User Registry. The procedure is slightly different, depending on how you perform the upgrade. If you see Figure 11 on page 116 you will see that there are three possibilities:

**Direct upgrade of existing master domain manager**

In this scenario the upgrade wizard or command attempts to reconfigure your authentication mechanism for the Federated User Registry. In the event that it fails to do this, but in all other respects the upgrade is successful, your upgraded master domain manager is configured with a stand-alone user registry. However, this is a temporary measure to allow you access to the master domain manager, and the problem must be resolved and the configuration migrated into the Federated User Registry before any further components are upgraded or installed.

**Parallel upgrade by installing a new backup master domain manager**

In this scenario, after you have installed the new backup master domain manager you must manually migrate your authentication mechanism from your old master domain manager to the new backup master domain manager. When the new backup master domain manager is switched to become the new master domain manager it is thus already configured for your current chosen authentication mechanism. The migration requires you to export the authentication configuration on the master domain manager to a text file and import it into the backup master domain manager. This is done using the WebSphere Application Server tools.

**Parallel upgrade by upgrading your existing backup master domain manager**

In this scenario your existing backup master domain manager will already be configured with the same user authentication as your master domain manager. The upgrade wizard or command attempts to reconfigure your authentication mechanism for the Federated User Registry. In the event that it fails to do this, but in all other respects the upgrade is successful, your upgraded backup master domain manager is configured with a stand-alone user registry. However, this is a temporary measure to allow you access to the backup master domain manager, and the problem must be resolved and the configuration migrated into the Federated User Registry before any further components are upgraded or installed.
RACF considerations:
In versions of Tivoli Workload Scheduler before V8.6, RACF authentication was configured using the IBM Tivoli Directory Server, and instructions about how to do this were given in the documentation. When you upgrade to V8.6 this configuration is also upgraded, and you can continue to use it. However, the Federated User Registry supports the z/OS Integrated Security Services LDAP Server, which you can configure after the upgrade is complete.

Mapping between stand-alone and federated repository keywords:
When you upgrade your stand-alone user registry, either by upgrading an existing component that uses WebSphere Application Server, or by exporting your configuration from the stand-alone to the federated environment, the product maps your existing configuration keywords to the new structure. It does so following these rules:
- If only old keywords are present, these keywords are used to configure the new keywords
- If both old and new keywords are present, the old keywords are discarded and only the new keywords are used.
- If the showSecurityProperties WebSphere Application Server tool is used to extract the security configuration based on the stand-alone repositories, a warning is issued to alert the user that, when re-imported, the data in the file is used to enable VMM security and the stand-alone repositories are disabled
- The LDAPreuseConnection property is not supported in federated repositories, and is discarded if present.
- The old LDAP user search filters, if specified, are converted to the new user search filters as follows. For more details about the individual properties discussed here see the Tivoli Workload Scheduler: Administration Guide.

LDAPUserFilter=(&(mail=%v)(cn=%v))(objectclass=ePerson)
All the parameters (like "mail=%v" and "cn=%v") are removed from the search filter and added to "LDAPloginProperties" (if not already present) separated by semi-colons ";" because these variables are not supported in the federated LDAP search filters. In this new configuration, VMM does not support replacement parameters such as "%v". In VMM, the filter to substitute mail with the specified value is applied by the VMM runtime during login to the application server, according to the login properties specified for an LDAP registry configured in the federated repository. The remaining values are added to the LDAPPUserSearchFilter. All values like objectclass=ePerson, objectclass=...... are added to the LDAPPUserObjectClasses separated with semi-colons ";" and without the keyword "objectclass ".

LDAPGroupFilter=(&(cn=%v)(ou=memberlist)(ou=ibmgroups)(o=ibm.com)(objectclass=groupOfUniqueNames))
All the parameters (like "cn=%v") are removed from the search filter and added to "LDAPloginProperties" (if not already present) separated with semi-colons ";". Because the variables are not supported in the federated LDAP search filters. In this new configuration, VMM does not support replacement parameters such as "%v". In VMM, the filter to substitute mail with the specified value is applied by the VMM runtime during login to the application server, according to the login properties specified for an LDAP registry configured in the federated repository. The remaining values are added to the LDAPPGroupSearchFilter. All values like (objectclass=groupOfUniqueNames , objectclass=......) are added to the LDAPPGroupObjectClasses separated with semi-colons ";" and without the keyword "objectclass ".

120 IBM Tivoli Workload Scheduler: Planning and Installation
LDAPUserIdMap=*:cn
If the first part of the mapping is not "*" and this value is present in the "LDAPloginProperties", then it is replaced in "LDAPloginProperties" using the second part of the mapping, in this case "cn".

LDAPGroupIdMap=*:cn
If the first part of the mapping is not "*" and this value is present in the "LDAPloginProperties", then it is replaced in "LDAPloginProperties" using the second part of the mapping, in this case "cn".

LDAPGroupMemberIdMap=groupOfNames:member;groupOfUniqueNames:uniqueMember
This sequence of a colon "\:" separated values is parsed and the first element of every pair of values is added to "LDAPGroupConfigMemberClasses", while the second element is added to "LDAPGroupConfigMemberNames".
- If the server type is IDS, then the "LDAPGroupConfigMemberScopes" is set to "all" and "LDAPGroupConfigName" is set to "ibm-allGroups"
- If the server type is AD, then the "LDAPGroupConfigMemberScopes" is set to "direct" and "LDAPGroupConfigName" is set to "memberOf"

EXAMPLE:

Stand-alone Configuration:
LDAPServerType=IBM_DIRECTORY_SERVER
LDAPUserFilter=((&(|(mail=%v)(cn=%v))(objectclass=ePerson))
LDAPGroupFilter=((&cn=%v)
  (ou=memberlist)
  (ou=ibmggroups)
  (o=ibm.com)
  (objectclass=groupOfUniqueNames))
LDAPUserIdMap=*:cn
LDAPGroupIdMap=*:cn
LDAPGroupMemberIdMap=groupOfNames:member;groupOfUniqueNames:uniqueMember

Federated Configuration
LDAPServerType=IDS
LDAPloginProperties=mail;cn
LDAPUserEntityType=PersonAccount
LDAPUserObjectClasses=ePerson
LDAPUserSearchFilter=(objectclass=ePerson)
LDAPGroupEntityType=Group
LDAPGroupObjectClasses=groupOfUniqueNames
LDAPGroupSearchFilter=((&ou=memberlist)
  (ou=ibmggroups)
  (o=ibm.com)
  (objectclass=groupOfUniqueNames))
LDAPGroupConfigName=ibm-allGroups
LDAPGroupConfigScope=all
LDAPGroupConfigMemberNames=member;uniqueMember
LDAPGroupConfigMemberClasses=groupOfNames;groupOfUniqueNames
LDAPGroupConfigMemberScopes=all;all
LDAPGroupConfigMemberDummies=uid=dummy;

changeSecurityProperties output:
The output from the changeSecurityProperties script, whether run by the upgrade wizard as part of the upgrade or run by you manually, gives a lot of information, warning, and error messages described in the section "changeSecurityProperties output" of the Tivoli Workload Scheduler: Administration Guide.

Performing a safe upgrade:
If you are upgrading in parallel mode you do not interrupt any running processes. However, if you are upgrading in direct mode you will interrupt running processes for the time it takes to perform the upgrade. To ensure that this interruption does not risk the integrity of these running processes, it is performed in safe mode. Safe mode ensures this by doing the following before starting the upgrade:

- Checks if there are command lines currently running.
- Prevents other jobs from starting by setting the job fence on the workstation to the go (101) value.
- Checks if there are jobs running. If there are, it waits 60 minutes and checks again. If all the jobs do not complete during this interval, the upgrade does not proceed and an error message is displayed. If you want to change this value, specify the number of minutes when you run the setup or perform a silent installation.
- Check if there are processes running. If any, it stops them, and waits for the completion of the stop action.

If all these checks are passed Tivoli Workload Scheduler starts the upgrade:

- If the upgrade complete successfully after the Batchman process restarts, the job fence is set to the original value, because there is a synchronization between the Batchman message queues and the Symphony file for the job fence value. The installation process does not start the Batchman process and to set the original job fence on your workstation, run:

  `conman "start"`

- If the upgrade does not complete successfully either because the checks are not passed or an error occurs, the job fence is not set to the original value. You must:
  - Set the job fence manually to its original value.
  - Perform the steps to complete the actions or correct the errors and resume the upgrade.

**Note:** The safe upgrade is available with all the upgrade methods except using Software Distribution.

### Preparing to upgrade

Before you begin the upgrade process, complete the following tasks as appropriate:

#### Configuring the security file for new functions

During the upgrade procedure you will need to configure the security file for new functions. Configuring the security file requires the following:

**Setting the default variable table when upgrading from version 8.3 and 8.4**

Setting variable tables applies to version 8.3 and higher. When you upgrade from version 8.3 and 8.4, the upgraded security file includes the new default statement for variable tables. A variable table is an object that groups together multiple variables. Known as global parameters in previous versions, these objects are now called variables in version 8.5 and higher. Local parameters are managed as before, and the old parameters statement from the previous security file continues to manage their security aspects. Any global parameters that were defined in your previous database have become elements of the default variable table, and a security statement for the table has been added to the security file. You can choose to customize the user permissions to this table (by default, all users have full permission).
Configuring for event-driven workload automation

Configuring for event-driven workload automation applies only to version 8.3. You must modify the security file to include new security statements for the event management and reporting features. If you have specific security settings in your V8.3 environment, these settings must be manually merged with the new settings before you build the final security file for your current environment. The statements you add manually might vary depending on your security settings and on whether you have chosen the parallel or direct upgrade scenario, as explained in the following sections.

Perform a backup of your database

Before you begin the upgrade process, perform a backup of your current Tivoli Workload Scheduler database, referring to the Oracle or DB2 documentation.

Upgrade your database to 64-bit

If you are performing a direct upgrade on any UNIX or Linux platform, except for Linux Intel 32 and HP RISC, you must upgrade your local DB2 or Oracle instances to 64-bit versions.

Linux kernel

If you are upgrading in a Linux environment that uses the LD_ASSUME_KERNEL=2.4.1 environment variable, upgrade to the current version of Tivoli Workload Scheduler in a shell that also uses the LD_ASSUME_KERNEL=2.4.1 environment variable.

Ensure your current Tivoli Workload Scheduler installations are in the correct state

When you are upgrading your current environment, make sure the software package is in the COMMIT state. If it is in the UNDOABLE state, you must accept it to change its state to COMMIT before you upgrade to the current version. To check the state, perform the following:

1. From <TWA_dir>/TWS/_uninstall/CLI, run the command: swd_env.bat or . . /swd_env.sh as appropriate.
2. Run the command: wdlssp
3. Search for the following lines:
   
   DISSE0164I Name : TWS_LP_twsuser
   DISSE0165I Version : 8.4.0.00
   DISSE0166I State : IC---
   ----------------------------------------
   DISSE0164I Name : FP_TWS_WINDOWS_twsuser
   DISSE0165I Version : 8.4.0.00
   DISSE0166I State : IC---
4. Ensure that the state of the package is IC.

Ensure your backup master domain manager is installed and working

If you do not use a backup master domain manager you are strongly recommended to install and use one during the upgrade process. Follow the instructions in the Tivoli Workload Scheduler: Administration Guide, SC23-9113 to configure the backup master domain manager and ensure that key files are regularly mirrored to it. Test that it is working correctly by running switchmgr to switch to it and back again.
Check for absolute installation paths

Use composer to check if the workstation has jobs or file dependencies that use the absolute installation path. If there are, modify the path using a variable.

New directory structure

This section applies if you are upgrading from version 8.3 or 8.4. If you are upgrading from version 8.5, this directory structure already exists. This section describes the new product directory structure and the new directory structure for SSL files that was implemented in version 8.5.

Product directory:
The new directory structure to all upgrades with the exception of upgrades performed using Software Distribution.

When you upgrade to the current version from version 8.3 or 8.4, a new product directory structure is created. During the upgrade process, Tivoli Workload Scheduler is moved from the old directory structure and then updated into the new directory structure. The new structure changes the existing TWShome to TWAhome which becomes the parent directory for the new TWShome.

UNIX and Linux operating systems

The product is installed in the user's home directory. The default location for the upgrade is:

- on Linux: /opt/ibm/TWA/TWS/
- on UNIX: /opt/IBM/TWA/TWS/

Windows operating systems

The default location for the upgrade is %ProgramFiles%\IBM\TWA\TWS\.

Note: The WebSphere Application Server located inside the installation directory is renamed from appserver to eWAS

Note:

1. If you have a FINAL schedule, during the upgrade, it is downloaded during the installation. The default FINAL is reused. A backup copy of the schedule is created with the name SFinal.extract in the new installation directory

2. If you have any custom configurations (for example, custom scripts or backup processes) existing in your Tivoli Workload Scheduler structure, you must update them so that they work in the new directory structure.

3. On UNIX operating systems, you can create symbolic links to the new directory structure until the scheduling environment is updated by performing the ln -s command from the old installation directory. For example:

   ln -s bin TWS/bin
   ln -s config TWS/config

For more information about installation paths, see “Instances of Tivoli Workload Automation” on page 27

Examples:

UNIX and Linux operating systems

If you originally installed Tivoli Workload Scheduler in
/export/home/twsuser, you have a directory structure as follows:
When you upgrade Tivoli Workload Scheduler, the new directory structure is:
/export/home/twuser/TWA/TWS/bin
/export/home/twuser/TWA/TWS/config
/export/home/twuser/TWA/TWS/uninstall
/export/home/twuser/TWA/eWAS
/export/home/twuser/TWA/wastools
/export/home/twuser/TWA/TDWB

**Windows operating systems**

If you originally installed Tivoli Workload Scheduler into the directory c:\Program Files\IBM\TWS, you have a directory structure as follows:
c:\Program Files\IBM\TWS\bin
c:\Program Files\IBM\TWS\config
c:\Program Files\IBM\TWS\uninstall
c:\Program Files\IBM\TWS\appserver
c:\Program Files\IBM\TWS\wastools

When you upgrade Tivoli Workload Scheduler, the new directory structure is:
c:\Program Files\IBM\TWA\TWS\bin
c:\Program Files\IBM\TWA\TWS\config
c:\Program Files\IBM\TWA\TWS\uninstall
c:\Program Files\IBM\TWA\eWAS
c:\Program Files\IBM\TWA\wastools
c:\Program Files\IBM\TWA\TDWB

**File system structure:**

When performing the upgrade the file system directory structure changes as follows:

**On UNIX and Linux operating systems:**

From Version 8.3 and 8.4

<TWS_home_directory>/appserver/profiles/twsprofile/config/cells/
DefaultNode/nodes/DefaultNode/servers/server1/

From Version 8.5.1

<TWS_home_directory>/eWAS/profiles/twaprofile/config/cells/
DefaultNode/nodes/DefaultNode/servers/twaserver

To Version 8.6

<TWS_home_directory>/eWAS/profiles/TIPProfile/config/cells/
TIPCell/nodes/TIPNode/servers/server1

**On Windows operating systems:**

From Version 8.3 and 8.4

<TWS_home_directory>\appserver\profiles\twsprofile\config\cells\
DefaultNode\nodes\DefaultNode\servers\server1

From Version 8.5.1

<TWS_home_directory>\eWAS\profiles\twaprofile\config\cells\
DefaultNode\nodes\DefaultNode\servers\twaserver

To Version 8.6

<TWS_home_directory>\eWAS\profiles\TIPProfile\config\cells\TIPCell\nodes\TIPNode\servers\server1
Where \( <TWS\_home\_directory> \) is the directory where you originally installed the product.

**Directory for SSL files:**
When you upgrade to the current version from version 8.3 or 8.4, a new directory for SSL files is created. The following describes the old and new directory structures depending on whether you have chosen the default installation path or have customized the installation path.

If you are using the default installation path, the TWSServerTrustFile.jks and TWSServerKeyFile.jks files are located as follows. Note that in these cases, the values of keyFileName and trustFileName in the security properties are already set to default.

- **Previous directory structure**
  - \( TWSInstallationPath\AppServer\profiles\twsprofile\etc\TWSServerTrustFile.jks \)
  - \( TWSInstallationPath\AppServer\profiles\twsprofile\etc\TWSServerKeyFile.jks \)

- **New directory structure**
  - \( TWSInstallationPath\eWAS\profiles\TIPProfile\etc\TWSServerTrustFile.jks \)
  - \( TWSInstallationPath\eWAS\profiles\TIPProfile\etc\TWSServerKeyFile.jks \)

If you are using a customized installation path, the TWSServerTrustFile.jks and TWSServerKeyFile.jks files are located as follows. The old keys are left in their original directories but are also copied to the new directory. The location parameters of WebSphere Application Server will be set to the default path which is \( $\{USER\_INSTALL\_ROOT\}/etc/KEYNAME \). Note that the values of keyFileName and trustFileName in the security properties are set to the default paths which are \( $\{USER\_INSTALL\_ROOT\}/etc/TWSServerKeyFile.jks \) and \( $\{USER\_INSTALL\_ROOT\}/etc/TWSServerTrustFile.jks \).

- **Previous directory structure**
  - \( CustomizedInstallationPath\TWSServerTrustFile.jks \)
  - \( CustomizedInstallationPath\TWSServerKeyFile.jks \)

- **New directory structure**
  - \( TWSInstallationPath\eWAS\profiles\TIPProfile\etc\TWSServerTrustFile.jks \)
  - \( TWSInstallationPath\eWAS\profiles\TIPProfile\etc\TWSServerKeyFile.jks \)

**Performing a direct upgrade**
This section describes how to upgrade your environment using a direct scenario. See "Direct upgrade scenario - minimizing the time to upgrade" on page 117 for an overview. It is divided into the following procedures:

- "Direct 1: Unlink the master domain manager from the network and stop it" on page 127
- "Direct 2: Upgrade the master domain manager or backup master" on page 128
- "Direct 3: Reconfigure the DB2 properties on the backup master domain manager" on page 132
- "Direct 4: Check the authentication upgrade was successful" on page 132
Direct 1: Unlink the master domain manager from the network and stop it:
Before commencing the upgrade, you must unlink all workstations from the master domain manager and stop it.

Follow these steps:
1. Log in as the `<TWS_user>`.
2. Unlink all workstations in the domain:
   - From the Dynamic Workload Console
     - From the Engine Selection Panel, select all workstations and click Unlink.
   - From the command line of the master domain manager
     - Issue the following command:
       `conman "unlink @;noask"`
3. Stop the master domain manager:
   - From the Dynamic Workload Console
     - From the Engine Selection Panel, select all workstations and click Stop.
   - From the command line of the master domain manager
     - Issue the following command:
       `conman "stop;wait"`
4. From the command line of the master domain manager, stop the netman process as follows:
   - UNIX and Linux operating systems
     - Run:
       `conman "shut" ; wait`
     - **Note:** Do not use the UNIX `kill` command to stop Tivoli Workload Scheduler processes.
   - Windows operating systems
     - Run the `shutdown.cmd` command from the Tivoli Workload Scheduler home directory.
5. If you are upgrading from version 8.4 or higher, stop the SSM Agent as follows:
   - UNIX and Linux operating systems
     - Run `conman stopmon` to stop the agent.
   - Windows operating systems
     - Stop the Windows service: Tivoli Workload Scheduler SSM Agent (for `<TWS_user>`).
6. Verify that all services and processes are not running, as follows:
   - UNIX and Linux operating systems
     - Run
ps -u <TWS_user>

Verify that the following processes are not running: netman, mailman, batchman, writer, jobman, JOBMAN, stageman, appserverman. All processes must be stopped with the exception of the embedded WebSphere Application Server which must remain running.

**Windows operating systems**

Run:
<drive>\unsupported\listproc.exe

where <drive> is the Tivoli Workload Scheduler home directory. Verify that the following processes are not running: netman, mailman, batchman, writer, jobman, stageman, JOBMON, tokensrv, batchup.

Also, ensure that no system programs are accessing the directory or subdirectories, including the command prompt, and that in Windows Explorer the **Administrative Tools>Services** panel is not open.

**Direct 2: Upgrade the master domain manager or backup master**

This section describes how to upgrade a master domain manager or backup master domain manager

You can upgrade a master domain manager or backup master domain manager using the following installation methods:

- “Upgrading using the installation wizard”
- “Upgrading using the silent installation” on page 132

**Upgrading using the installation wizard:**

To upgrade a Tivoli Workload Scheduler master domain manager or backup master domain manager perform the following steps.

1. Choose one of the following:
   - Insert the installation DVD and run the setup for your operating system:
     - **On UNIX and Linux operating systems**
       TWS/operating_system/SETUP.bin or TWS/SETUP.sh
     - **On Windows operating systems**
       TWS\operating_system\SETUP.exe or TWS\SETUP.cmd
   - Alternatively, start the launchpad as follows and select the Tivoli Workload Scheduler installation:
     - **On UNIX and Linux operating systems**
       From the root directory of the DVD, run launchpad.sh.
     - **On Windows operating systems**
       From the root directory of the DVD, run launchpad.exe.
       If you have autorun enabled, the launchpad starts automatically. If you want to start the launchpad from a mounted file system, ensure that you have write permission on it before starting the launchpad.

2. Follow the installation wizard screens to complete the installation. The following list describes the fields that you might need to complete during the installation. Some fields might not apply to your upgrade depending on whether your old version is 8.3 or higher.

**Back up the previous Tivoli Workload Scheduler instance**

Select whether to back up the previous instance.
**Backup Destination Directory**
When you select to back up the previous instance, specify the directory where the backup is to be located.

**Backup profile destination directory**
The upgrade procedure performs a backup of your WebSphere® Application Server (WAS) profile. Your current settings are transferred to the new WebSphere Application Server automatically. A default backup path is provided for you. If you want to save the profile to a different path, specify it in this field. Click Next.

**Tivoli Workload Scheduler user password**
Type the password of the Tivoli Workload Scheduler user for which you are upgrading the instance.

**Embedded WebSphere Application Server (WAS) administrator user name and password**
If you have changed the WebSphere Application Server (WAS) administrator user name and password from your previous installation, enter the new user name and password. If you have not changed these WebSphere Application Server values in your current installation, leave these fields blank.

**SAS Server Authentication Listener**
Applies only to version 8.3. The port used by the Secure Association Services (SAS) to listen for inbound authentication requests. The default value is 31119.

**CSIv2 Server Authentication Listener**
Applies only to version 8.3. The port on which the Common Secure Interoperability Version 2 (CSIv2) service listens for inbound server authentication requests. The default value is 31120.

**CSIv2 Client Authentication Listener**
Applies only to version 8.3. The port on which the Common Secure Interoperability Version 2 (CSIv2) service listens for inbound client authentication requests. The default value is 31121.

**ORB Listener**
Applies only to version 8.3. The port used for RMI over IIOP communication. The default value is 31122.

**Administration HTTP transport**
Applies only to version 8.3. The administrative console port. The default value is 31123.

**Administration HTTPS transport**
Applies only to version 8.3. The administrative console secure port. The default value is 31124.

**Event Processor**
Applies only to version 8.3. This port is used by the event management feature. The default value is 31131. This port is not requested for backup master domain managers.

**JobManager port number**
The port used by the Tivoli Workload Scheduler for z/OS server or the dynamic workload broker component to connect to the Tivoli Workload Scheduler agent. It is used by JobManager to run dynamic workload and to run workload coming from a z/OS environment in a distributed environment. JobManager is the network process that controls the
dynamic scheduling environment and the z-centric environment. The
default value is 31114. The valid range is from 1 to 65535. This port
number is required only if you are upgrading the master domain
manager.

Host name or IP address
The fully qualified hostname on which the agent will be contacted by
the dynamic workload broker.

Enable the dynamic scheduling capability
Selecting this option enables the dynamic workload broker for the
dynamic scheduling capability of the master domain manager. If you
are upgrading a master domain manager, the Dynamic Workload
Broker workstation definition is created. For the dynamic scheduling
capability to be enabled, all the backup master domain managers in
your network must have installed Tivoli Workload Scheduler, version
8.3 Fix Pack 8 or later, or Tivoli Workload Scheduler, version 8.4 Fix
Pack 4 or later.

Although dynamic workload broker is not installed on the backup
master domain manager, dynamic scheduling is enabled and the related
WebSphere Application Server configuration tools are installed during
both the installation or upgrade. If you decide to enable dynamic
scheduling capability later see the procedure described in “Enabling
dynamic scheduling after installation” on page 168.

Dynamic workload broker workstation name
The definition of the dynamic workload broker workstation created in
the Tivoli Workload Scheduler database. Spaces are not allowed and the
maximum field length is 16 characters. It can contain alphanumeric,
dash (-), and underscore (_) characters. The first character must be a
letter.

The dynamic workload broker workstation acts as the communication
bridge between the master domain manager and the dynamic workload
broker component. In your job or job stream definitions it is the
workstation where the jobs run. In this way, you submit your workload
through this workstation to the dynamic workload broker component.

Dynamic workload broker host name
Applies to backup master domain manager. The dynamic workload
broker fully qualified host name. Adds the capabilities to run dynamic
workload to the Tivoli Workload Scheduler agent. If not specified, the
default value is localhost. This value is registered in the
ResourceAdvisorUrl property in the JobManager.ini file.

Dynamic workload broker Netman port
Applies to backup master domain manager. The dynamic workload
broker Netman port number used to add dynamic scheduling
capabilities to your distributed or end-to-end environment. This
number is registered in the ResourceAdvisorUrl property in the
JobManager.ini file. The default value is 31116. The valid range is from
0 to 65535. If you specify 0, you do not add the capability to run
dynamic workload to the agent.

RDBMS installation path
Depending on the type of RDBMS you are using, specify the following
information:

For DB2:
**DB2 Server administrator user**

The user name of the administrator of the DB2 server instance. This user can also be any user having SYSADM or SYSCTRL authority on the DB2 server. On UNIX, verify that you are able to switch to this user and that it can load the DB2 environment.

If the DB2 administrator already created the database tables using the "Creating or upgrading the database tables if you are using DB2" on page 37 procedure, the user name is the one that the DB2 administrator specified in the **DB_USER** property in the `customizeDB2SQL.properties` file.

**DB2 Server administrator password**

The password of the DB2 server administrator user, or of the user with SYSADM or SYSCTRL authority. You are asked to confirm the password.

**Note:** The DB2 installation path will be discovered automatically by the upgrade procedure.

If you have a DB2 Enterprise client installed on a UNIX platform, specify also the **DB2 local client user name**.

For Oracle:

**Installation path**

Specify the path of an Oracle installation that satisfies the Tivoli Workload Scheduler prerequisites. The fully qualified path must identify a tree in the Oracle structure that includes the `sqlplus` executable.

**Oracle Administrator User**

The name of the Oracle Administrator user.

If the **ORACLE administrator already created the database tables using the "Creating or upgrading the database tables if you are using Oracle" on page 47** procedure, the user name is the one that the ORACLE administrator specified in the **MDL_USER** property of the `customizeORACLESQL.properties` file.

**Oracle Administrator user password**

The password of the Oracle Administrator user. You are asked to confirm the password.

**Tivoli Workload Scheduler database information**

Applies only when upgrading from version 8.3. Specify the following information needed to update the Tivoli Workload Scheduler database:

For DB2:

**Report tablespace name**

The name of the tablespace used to store event logs

**Report tablespace path**

The path of the tablespace used to store event logs.

**Note:** If you are upgrading from Tivoli Workload Scheduler version 8.3 fix pack 1 or higher, the DB2 installation path will be discovered automatically by the upgrade procedure.

For Oracle:
Installation path
Specify the path of an Oracle installation that satisfies the Tivoli Workload Scheduler prerequisites. The fully qualified path must identify a tree in the Oracle structure that includes the sqlplus executable.

Tivoli Workload Scheduler Oracle user password
The password of the Tivoli Workload Scheduler Oracle user

Create the Tivoli Workload Scheduler schema using the Oracle Partitioning option
If you are upgrading version 8.3 on Oracle Enterprise Edition and have not implemented the Oracle Partitioning feature, you can do so at this point. Implementing this feature improves the performance of event-driven workload automation. Note that the partitioning option must already be installed into the Oracle instance. For information about event-driven workload automation, see Overview.

If you are upgrading version 8.4 and higher, this option is not available because the database schema of the event-driven workload automation already exists. To implement the Oracle Partitioning feature, see Administration Guide.

Tivoli Workload Scheduler report tablespace
The path of the Oracle tablespace for Tivoli Workload Scheduler reports.

Upgrading using the silent installation:
To upgrade your Tivoli Workload Scheduler master domain manager or backup master domain manager instance, use one of the following response files:
- TWS86_UPGRADE_MDM_83plus_UNIX.txt
- TWS86_UPGRADE_BACKUP_MDM_83plus_UNIX.txt
- TWS86_UPGRADE_MDM_83plus_WIN.txt
- TWS86_UPGRADE_BACKUP_MDM_83plus_WIN.txt
and follow the procedure described in “Performing a silent installation” on page 84.

Direct 3: Reconfigure the DB2 properties on the backup master domain manager:
If you are upgrading a backup master domain manager and using either a DB2 client or server, you must perform the following step after completing the upgrade procedure.

Open the TWA_HOME\TDWB\config\CLIConfig.properties file. In this file, edit the following property:
com.ibm.tdw.db.dao.rodbms.jdbcPath=jdbc:db2://DB2_server_hostname://DB2_server_port/DB_name

where:
- DB2_server_hostname is the hostname of the DB2 server.
- DB2_server_port is the port number of the DB2 server.
- DB_name is the name of the database. This field will be already customized during the upgrade.

Direct 4: Check the authentication upgrade was successful:
The upgrade attempts to migrate your authentication configuration (for example, Local OS, LDAP, or File Registry) to the new format inside a Federated User Registry (see "Updating authentication" on page 118 for an overview of the new authentication features and how the upgrade is performed). It is possible that this migration might fail, in which case the following happens:

**Upgrade status**

If there are no other upgrading problems the upgrade will finish as "successful", even though this aspect of it has failed.

**Authentication configuration**

Your existing authentication configuration will be implemented as "stand-alone". This is a temporary state which must be corrected before attempting to install or upgrade other components that need to communicate with the master domain manager. To do this, follow the instructions in the *Tivoli Workload Scheduler: Administration Guide* about how to configure authentication in a Federated User Registry, using either the Integrated Solutions Console or the WebSphere Application Server tools.

**Installing the Dynamic Workload Console**

If you attempt to install the Dynamic Workload Console in the same instance of Tivoli Workload Automation as the master domain manager it will fail because the user registry is not federated. If you need to install the Dynamic Workload Console temporarily, install it in a new instance of Dynamic Workload Console. However, you should note that this is not the most efficient way to use the Dynamic Workload Console and you should plan to remove it and reinstall it in the same Tivoli Workload Automation instance as the master domain manager as soon as the Federated User Registry is configured and working.

To determine if the migration of the authentication configuration has failed you must consult the upgrade log (see "Installation log files" on page 32 for information about where to find the log.

**Direct 5: Customize and submit the optional final job stream:**

If your old final job stream is called FINAL, a backup copy has been made of it in Sfinal.extract and it has been upgraded to V8.5.1. If it was customized, you must make the corresponding customizations to the new FINAL job stream. If it is not called FINAL, you must merge the functions of your old final job stream with the syntax of your new FINAL job stream. Depending on your situation, perform the following steps:

1. Customize the final job stream as required:
   - **If you had a customized job stream called FINAL in your database:**
     a. Edit the new FINAL job stream with composer or the Dynamic Workload Console.
     b. Edit the file $final.extract with a text editor.
     c. Make the corresponding customizations to the new FINAL job stream.
     d. Save your new FINAL job stream.
   - **If you had a customized final job stream called something other than FINAL in your database:**
     a. Edit the new FINAL job stream with composer or the Dynamic Workload Console.
     b. Edit your old final job stream with composer or the Dynamic Workload Console.
c. Merge the two job streams so that your new final job stream has the same name and customizations as before (if you want to preserve the naming), plus the new required attributes from the new FINAL job stream.

d. Save your new final job stream.

e. Delete the old final job stream.

If you had a final job stream called something other than FINAL in your database, but it is not customized:

a. Delete your old final job stream with composer or the Dynamic Workload Console.

b. Rename the new FINAL job stream with the name of your old final job stream with composer or the Dynamic Workload Console.

If you had a final job stream called FINAL in your database, but it is not customized:

Take no action because the FINAL job stream has already been edited by the installation or upgrade procedure.

2. Use conman to delete your old final job stream instances and submit new instances to replace them.

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

Note: You run JnextPlan in "Direct 7: Complete the security configuration for the new environment" on page 135.

Direct 6: Upgrade your backup master domain manager:

If you use a backup master domain manager you must now upgrade it to the same version as the master domain manager, otherwise the new functions are not supported. Much of what you have to do is the same as for the master domain manager. Follow these instructions:

- "Direct 2: Upgrade the master domain manager or backup master" on page 128
- "Direct 3: Reconfigure the DB2 properties on the backup master domain manager" on page 132
- "Direct 4: Check the authentication upgrade was successful" on page 132. You should have implemented the same authentication for your backup master domain manager as your master domain manager, so you should expect to get the same results from the authentication upgrade, but even if the master domain manager authentication upgrade completed successfully, you must still check the upgrade log for the backup master domain manager.
- There is no need to make any changes to the FINAL job stream because the FINAL job stream on the master domain manager is used whenever you run the switch manager process.
- Neither do you need to change the security file configuration on the backup master domain manager, because the procedure for maintaining your environment in readiness for the use of switch manager require you to mirror the Security file to the backup master domain manager whenever you change it.
If you do not use a backup master domain manager, you are strongly recommended to install and use one, to ensure the high availability of your scheduling environment.

**Direct 7: Complete the security configuration for the new environment:**

Versions 8.5 and higher include new security statements for the event management and variable tables. If you have specific security settings in your V8.3 or V8.4 environment, these settings must be manually merged with the new settings before you build the final security file to be used in your new environment. The statements you might have to add manually vary depending on your specific security settings.

Perform the following:

1. Log in as `<TWS_user>` on your upgraded master domain manager and set the Tivoli Workload Scheduler environment.
2. If you have centralized security enabled, extract the new security file on the new master using the following V8.5 and higher command:
   ```
dumpsec > sec_file
   ```
   where `sec_file` is the text file created by the `dumpsec` command.
3. Edit the `sec_file`, and insert the following statements:
   If you are upgrading from V8.3, add the following statements:
   ```
   REPORT NAME=@ ACCESS=DISPLAY
   EVENTRULE NAME=@ ACCESS=ADD,DELETE,DISPLAY,MODIFY,LIST,UNLOCK
   ACTION PROVIDER=@ ACCESS=DISPLAY,SUBMIT,USE,LIST
   EVENT PROVIDER=@ ACCESS=USE
   ```
   If you are upgrading from a V8.3 or V8.4 environment, add the following statement:
   ```
   VARTABLE NAME=@ ACCESS=ADD,DELETE,DISPLAY,MODIFY,USE,LIST,UNLOCK
   ```
4. Check that the user permissions of the new statements are correct.
5. Save your changes to the `sec_file`.
6. Build your final security file for your new master domain manager using the V8.5 and higher `makesec` command:
   ```
makesec sec_file
   ```
7. If you are using FIPS, you must manually enable it again in the WebSphere Application Server java.security file. See the *Tivoli Workload Scheduler: Administration Guide* for the FIPS compliance information.
8. If you have centralized security, distribute the security file.
9. Run `JnextPlan -for 0000` to distribute the Symphony file to the agents.

   **Note:** Ensure that the `optman cf` option is set to `all` or only the unfinished jobstreams will be carried forward.
10. Restore the previous setting of the `optman cf` option, if necessary.
11. If you want to use EDWA, enable it using `optman`.

**Direct 8: Restart scheduling processes:**

Restart the scheduling processes after the upgrade is complete, as follows:

1. Log in as the `<TWS_user>`.
2. Start the master domain manager:
   ```
   From the Dynamic Workload Console
   From the Engine Selection Panel, select all workstations and click **Start**.
   ```
From the command line of the master domain manager
   Issue the following commands:
   conman "startappserver";wait
   conman "start"

3. From the command line of the master domain manager, start the netman process as follows:
   UNIX and Linux operating systems
   Run:
   StartUp.sh

   Windows operating systems
   Run:
   StartUp

4. Run
   conman start

5. Link all workstations in the domain:
   From the Dynamic Workload Console
   From the Engine Selection Panel, select all workstations and click Link.

   From the command line of the master domain manager
   Issue the following commands:
   conman "link @;noask"

6. If you want your upgraded environment to perform event processing, firstly, run:
   conman startevtp

   Then do the following:
   From the Dynamic Workload Console
   a. Click Tivoli Workload Scheduler>Scheduling Environment>Monitor>Monitor Workstations
   b. Select All Workstations in plan or another predefined task name
   c. Choose an engine name, or specify connection properties, and click OK
   d. Select a workstation and click More Actions>Start Event Monitoring.

   From the command line of the master domain manager
   UNIX and Linux operating systems
   Run
   conman startmon

   Windows operating systems
   Start the Windows service: Tivoli Workload Scheduler SSM Agent (for <TWS_user>).

7. Verify that all services and processes are running, as follows:
   UNIX and Linux operating systems
   Run
   ps -u <TWS_user>

   Verify that the following processes are running: netman, mailman, batchman, writer, jobman, JOBMAN, stageman, appserverman.
Windows operating systems

Run:

<drive>\unsupported\listproc.exe

where <drive> is the Tivoli Workload Scheduler home directory. Verify that the following processes are running: netman, mailman, batchman, writer, jobman, stageman, JOBMON, tokensrv, batchup.

Note: Even if the autotrace mechanism is no longer supported, the upgrade process does not remove the TWA_home\TWS\trace directory after the upgrade because it is possible that you use it with other Tivoli products. If you are sure that you do not use it, you can remove the TWA_home\TWS\trace directory.

Performing a parallel upgrade

This section describes how to upgrade your environment using a parallel upgrade scenario. See “Parallel upgrade scenario - minimizing the impact on scheduling” on page 117 for an overview.

The scenario consists of the following procedures:

- “Parallel 1: Upgrade your existing backup master domain manager, or install a new one”
  - “Parallel 1a: Install a new backup master domain manager”
  - “Parallel 1b: Upgrade your current backup master domain manager” on page 138
- “Parallel 2: Switch the master domain manager to the new or upgraded backup master” on page 139
- “Parallel 3: Make the switch manager permanent” on page 139
- “Parallel 5: Install a new master domain manager or upgrade your old master domain manager” on page 140
- “Parallel 6: Switching back to the old master domain manager (optional)” on page 141
- “Parallel 7: Complete the security configuration for the new environment” on page 141

The upgrade process changes some files and folders, see “Files and folders changed during the upgrade” on page 113 for the complete list.

Parallel 1: Upgrade your existing backup master domain manager, or install a new one:

This step is divided into two alternative substeps, depending on whether you already have a backup master domain manager in your environment:

Parallel 1a: Install a new backup master domain manager:

The purpose of this step is to install a fresh backup master domain manager and attach it to your current network.

This backup master domain manager will point to your existing Tivoli Workload Scheduler database and become your new master domain manager.

Parallel 1a-1: Install the fresh backup master domain manager

To install a new backup master domain manager see the procedures described in “Installing” on page 59. Specifically, see the procedure described in “Tivoli Workload Scheduler installation options” on page 61.
and subsequent sections depending on whether you are using a DB2 or an Oracle database. Ensure that your new backup master domain manager points to your current Tivoli Workload Scheduler database instance.

**Parallel 1a-2: Migrate your authentication configuration**

Do the following to migrate your authentication mechanism to the newly installed backup master domain manager:

1. On your existing master domain manager, use the `showSecurityProperties` tool to export your authentication configuration to a text file.
2. Copy this file to your new backup master domain manager.
3. Run the `changeSecurityProperties` tool on the new backup master domain manager to import the configuration. The tool recognizes that the input file is in the old format and attempts to migrate the configuration to the new format.
   
   If your authentication mechanism is customized in ways that the migration cannot handle, an error or errors will be given and you should configure the authentication mechanism manually.
4. Test that the migrated authentication mechanism allows you to log on and use `composer` with more than one user ID.

**Parallel 1a-3: Define a new backup master domain manager in the database**

Define your new backup master domain manager as a full status agent in the domain of your master domain manager, using the `composer` command interface.

**Parallel 1a-4: Optionally enable dynamic scheduling capabilities**

The upgrade installs a disabled dynamic workload broker on the backup master domain manager. If you do not want dynamic scheduling capabilities in your network, you do not have to take action here. You can enable dynamic scheduling capability later following the procedure described in "Enabling dynamic scheduling after installation" on page 168.

**Parallel 1a-5: Prepare the old security file for switching the manager**

For Tivoli Workload Scheduler to switch correctly, you must add the new `TWS_user` into the old security file. The new `TWS_user` is the one that you used when you installed the new backup master domain manager.

Perform the following steps:

1. On the old master domain manager, log in as the old `TWS_user` and set the Tivoli Workload Scheduler environment. Add the `TWS_user` of the new master domain manager to the old security file.
2. If you have centralized security, distribute the security file to all agents. If you do not have centralized security, copy the compiled security file to the installed or upgraded backup master domain manager, overwriting the version that is there.

You now need to make these changes effective, as follows:

1. Ensure that the `optman cf` option is set to `all`
2. Run `JnextPlan -for 0000` or wait until the end of the production plan. This distributes the Symphony file to the new backup master domain manager.
3. Restore the previous setting of the `optman cf` option, if necessary.

**Parallel 1b: Upgrade your current backup master domain manager**
To upgrade your current backup master domain manager, follow the procedure described in “Direct 2: Upgrade the master domain manager or backup master” on page 128 using your preferred installation method.

**Parallel 2: Switch the master domain manager to the new or upgraded backup master:**
Switch to your new backup master domain manager, which now becomes your master domain manager, by issuing the following command on your old master domain manager:

```bash
conman
switchmgr masterdm;new_mgr_cpu
```

where `new_mgr_cpu` is the name of the workstation of your new or upgraded backup master domain manager.

If you are upgrading from V8.4 or higher and the event processor is being hosted on the old master domain manager or backup master domain manager, you must run `switchevtprocessor` to switch the event processor in the same way.

If using the back up master domain manager V8.6 you define agent, pool, or dynamic pool workstations and than you open their database definitions from the master domain manager V8.4 database, their workstation types are blank.

**Parallel 3: Make the switch manager permanent:**
In the preceding step you have promoted your upgraded backup master domain manager to the role of master domain manager.

To make this configuration fully operational and persistent through JnextPlan, you must perform the following steps:

On the new master domain 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 master. See the *Tivoli Workload Scheduler: Administration Guide*.

2. Change the workstation definition of the old master using `composer`:
   ```
   modify cpu=old_mgr_cpu
   ```

   and substitute `type=manager` with `type=fta`

3. Repeat the preceding step, this time to modify the workstation definition of the new master and substitute `type=fta` with `type=manager`.

4. Verify the plan date on the new master:
   ```
   planman showinfo
   ```

   Only if the output of the `planman showinfo` contains information, run the following command to remove the preproduction plan:
   ```
   planman reset -scratch
   ```

   For more information on `planman reset -scratch`, see *Tivoli Workload Scheduler: User’s Guide and Reference*. 

---

**Part 2: Tivoli Workload Scheduler**  
**139**
**Note:** This command removes the preproduction plan and maintains the Symphony file, while the `ResetPlan -scratch` command removes both, so it is important to use them appropriately.

5. Ensure that the `optman cf` option is set to `all`.
6. Rebuild the plan to activate the changes to the database:
   ```
   JnextPlan - from <start_date> -for 0000
   ```
7. Restore the previous setting of the `optman cf` option, if necessary.
8. Edit the `\TWS\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. See the Tivoli Workload Scheduler: Administration Guide.
   In this way the reports (`reptr - pre` and `reptr -post`) can run when you run `JnextPlan`.

**Note:** Ensure that the global option `carryforward` is set to `all` or only the unfinished jobstreams will be carried forward.

**Parallel 4: Customize the optional final job stream:**
Perform the procedure described in [Direct 5: Customize and submit the optional final job stream](#) on page 133.

**Parallel 5: Install a new master domain manager or upgrade your old master domain manager:**
You can now choose to install a new master domain manager, following the instructions in [“Installing a master domain manager or backup master”](#) on page 60.

Alternatively, you might decide to upgrade rather than install a new master domain manager.

During the upgrade you will be prompted to activate the dynamic scheduling capabilities on that computer. If you want to activate dynamic scheduling capabilities, you must be sure that ALL backup domain managers on the network are at the minimum supported version level. See the Tivoli Workload Scheduler System Requirements Document at [http://www.ibm.com/support/docview.wss?rs=672&uid=swg27019747](http://www.ibm.com/support/docview.wss?rs=672&uid=swg27019747) You can then choose to activate dynamic scheduling capabilities.

If you do not know if all backup master domain managers are at the minimum supported level, do not select the option to activate dynamic scheduling capabilities. You can add this capability at a later time using the “Enabling dynamic scheduling after installation” on page 168 procedure.

Before performing this step, if you created agent, pool, or dynamic pool workstations in Step 2 “Parallel 2: Switch the master domain manager to the new or upgraded backup master” on page 139 set them to `ignore`. If you do not set them to `ignore`, when the master domain manager adds the workstation definition to the plan it does not find them and sends a lot of messages in the `IBM\TWA\TWS\pobox` files. The size of these files increases exponentially.

To upgrade your master domain manager (which is now your backup master domain manager), perform the following steps:
1. From the new master domain manager, unlink the old master workstation
   conman "unlink old_mdm_cpu"

2. Upgrade your old master domain manager to the current version using the
   procedure described in “Direct 6: Upgrade your backup master domain
   manager” on page 134.

3. Link the upgraded master domain manager to the network
   conman "link old_mdm_cpu"

Parallel 6: Switching back to the old master domain manager (optional):
This step is optional. You can switch back to your old master domain manager that
has now been upgraded. To do this, perform the following steps:
1. From the upgraded master domain manager switch the master domain
   manager:
   conman
   switchmgr masterdm;old_mdm_cpu

2. To restore your upgraded master domain manager to its role permanently,
   perform the steps outlined in “Parallel 3: Make the switch manager permanent”
   on page 139, this time for the master workstation.

Parallel 7: Complete the security configuration for the new environment:
Perform the procedure described in “Direct 7: Complete the security configuration
for the new environment” on page 135.

Upgrading agents and domain managers
This section describes how to upgrade Tivoli Workload Scheduler agents and
domain managers in your distributed, z/OS, or end-to-end network. During the
upgrade, you can add dynamic scheduling capabilities or the Java runtime to run
job types with advanced options to the agent. The runtime environment is used to:
• Run job types with advanced options, both those supplied with the product and
  the additional types implemented through the custom plug-ins on the agent.
• Enable the capability to remotely run, from the agent, the dynamic workload
  broker resource command on the server.

See “Software packages and parameters” on page 97 for a description of the
parameters.

If you are upgrading from a version prior to 8.5.1 you can add new features
during the upgrade process.

If you are upgrading from version 8.5.1 and during the installation you did not
install some features like the dynamic capabilities or the Java runtime to run job
types with advanced options, you cannot add them during the upgrade process. To
add them perform the procedure described in the following sections:
• “Adding a feature” on page 165
• “Enabling dynamic scheduling after installation” on page 168

The product performs the upgrade in safe mode by performing all the checks
detailed in “Performing a safe upgrade” on page 121 before starting. To ensure that
the upgrade can run without stopping, perform manually the steps described in
“Unlinking and stopping Tivoli Workload Scheduler when upgrading agent
workstations” on page 142 before starting the upgrade.

The upgrade process changes some files and folders, see “Files and folders
changed during the upgrade” on page 113 for the complete list.
When the upgrade procedure is successful, it is not possible to roll back to the previous version. The following describes the new directory structure and how to upgrade agents using the various installation methods:

- **“New directory structure”**
- **“Unlinking and stopping Tivoli Workload Scheduler when upgrading agent workstations”**
- **“Upgrading agents and domain manager using the installation wizard” on page 143**
- **“Upgrading agents and domain manager using a silent installation” on page 145**
- **“Upgrading agents and domain manager using twsinst” on page 146**
- **“Upgrading agents using Software Distribution” on page 151**

**New directory structure**

For the new product directory structure and the new directory structure for SSL files see **“New directory structure” on page 124**.

**Unlinking and stopping Tivoli Workload Scheduler when upgrading agent workstations**

The product performs the upgrade in safe mode by performing all the checks detailed in **“Performing a safe upgrade” on page 121** before starting. To ensure that the upgrade can run without stopping, perform manually the steps indicated in the procedure before starting the upgrade.

Before you perform an upgrade on an agent workstation ensure that all Tivoli Workload Scheduler processes and services are stopped. If you have jobs that are currently running, the related processes must be stopped manually or you must wait until the jobs are complete.

**Note:** Do not use the UNIX kill command to stop Tivoli Workload Scheduler processes.

To stop Tivoli Workload Scheduler processes and services, follow these steps:

1. Unlink the target workstation from the other workstations in the network. Or, from the command line of the master domain manager, enter the following command:
   ```
   conman "unlink workstationname;noask"
   ```

2. Stop the target workstation. Or, from the command line of the master domain manager, log in as TWS_user and enter the following command:
   ```
   conman "stop workstationname;wait"
   ```

3. If you are upgrading from version 8.4 or higher, stop the SSM Agent as follows:
   - On Windows, stop the Windows service: Tivoli Workload Scheduler SSM Agent (for TWS_user).
   - On UNIX, run `stopmon` to stop the agent.

4. Stop the netman process as follows:
   - On Windows, from the Tivoli Workload Scheduler home directory, run the `shutdown.cmd`.
   - On UNIX, run:
     ```
     conman "shut;wait workstationname"
     ```

5. If you are updating an agent, remove (unmount) any NTFS mounted directories from the master domain manager.

To verify if any services and processes are still running, perform the following steps:
• On Windows operating systems, enter the command:
  `<drive>\unsupported\listproc.exe`

  Verify that the following processes are not running: netman, mailman, batchman,
  writer, jobman, stageman, JOBMON, tokensrv, batchup.
  Also, ensure that there are no system programs accessing the directory or
  subdirectories, including the command prompt. In Windows Explorer, the
  Administrative Tools→Services panel must be closed.

  Note:
  1. If you are upgrading in a Windows environment, the Tivoli Token
     Server must be running,
  2. Before you upgrade, make sure that the `conman` command line is not
     running

• On UNIX, enter the command:
  `ps -u TWS_user`

**Upgrading agents and domain manager using the installation wizard**

Use the installation wizard if you want to upgrade agents and satisfy the following
objectives:

**Perform the upgrade in a safe way**

  It checks all the processes that are running before starting. It does not
  perform the upgrade if there are command lines currently running and
  advises you if there are jobs running. In this case you can decide to wait
  before performing the upgrade or quit the upgrade. See “Performing a safe
  upgrade” on page 121 for detailed information.

**Upgrade the master, the backup master, and the agents using the same upgrade method**

  It can be run on all the workstations of your network.

**Use a graphical interface that guides the user through the upgrade**

  In interactive mode, the wizard guides you through the upgrade steps.

**Manage both UNIX and Windows operating system workstations**

  It runs both on UNIX and Windows agents.

To upgrade an agent using the installation wizard, run the setup for the operating
system on which you are upgrading:

**Windows operating system**

  From the WINDOWS directory, run:
  ```
  SETUP.exe
  ```

**UNIX and Linux**

  From the `operating_system` directory, run:
  ```
  SETUP.bin
  ```

  Note: The number of minutes that the product waits for jobs that are running to
  complete before starting the upgrade is 60. If the jobs do not complete
during this interval the upgrade does not proceed and an error message is
displayed. If you want to change this value run the following command
before starting the upgrade:

**Windows operating system**

  From the WINDOWS directory, run:
SETUP.exe -W checkJobsLoop.wait=value

**UNIX and Linux operating systems**
From the `operating_system` directory, run:
SETUP.bin -W checkJobsLoop.wait=value

where `value` is an integer or -1 for the product to wait indefinitely.

Alternatively, start the launchpad as follows and select the Tivoli Workload Scheduler installation:

**Windows operating system**
From the root directory of the DVD, run `launchpad.exe`.

**UNIX and Linux operating systems**
From the root directory of the DVD, run `launchpad.sh`.

When the installation wizard is launched, select the agent you want to upgrade and follow the prompts to complete the upgrade.

For a description of the options requested during the upgrade, see "Installation options" on page 80.

**Upgrading an agent with connector using the installation wizard:**
To upgrade a Tivoli Workload Scheduler agent version 8.3 or higher with a connector using the installation wizard, run the setup for the operating system on which you are upgrading:

**On Windows operating system**
TWS\`operating_system\`SETUP.exe or TWS\SETUP.cmd

**On UNIX and Linux operating systems**
TWS/`operating_system/SETUP.bin or TWS/SETUP.sh

Alternatively, start the launchpad as follows and select the Tivoli Workload Scheduler installation:

**Windows operating system**
From the root directory of the DVD, run `launchpad.exe`.

**UNIX and Linux operating systems**
From the root directory of the DVD, run `launchpad.sh`.

If you have autorun enabled, the launchpad starts automatically. If you want to start the launchpad from a mounted file system, ensure that you have write permission on it before starting the launchpad.

**Note:** During the upgrade, you are prompted for the WebSphere Application Server administration user name and password.

Follow the installation wizard panels to complete the upgrade. For a description of the agent options requested during the upgrade, see "Installation options" on page 80.

The following list describes the fields to complete for the connector:

**Backup profile destination directory**
This information is needed to perform a backup of your WebSphere Application Server (WAS) profile. Your current settings are transferred to WebSphere Application Server automatically.
Tivoli Workload Scheduler user password
Enter the password of the Tivoli Workload Scheduler user for which you are upgrading the agent and connector instance. If you changed the WebSphere Application Server administrator user name and password from your previous installation, you must supply them here. If you did not change these values in your installation, leave these fields blank.

SAS Server Authentication Listener
Applies only to version 8.3. The port used by the Secure Association Services (SAS) to listen for inbound authentication requests. The default value is 31119.

CSIv2 Server Authentication Listener
Applies only to version 8.3. The port on which the Common Secure Interoperability Version 2 (CSIv2) service listens for inbound server authentication requests. The default value is 31120.

CSIv2 Client Authentication Listener
Applies only to version 8.3. The port on which the Common Secure Interoperability Version 2 (CSIv2) service listens for inbound client authentication requests. The default value is 31121.

ORB Listener
Applies only to version 8.3. The port used for RMI over IIOP communication. The default value is 31122.

Administration HTTP transport
Applies only to version 8.3. The administrative console port. The default value is 31123.

Administration HTTPS transport
Applies only to version 8.3. The administrative console secure port. The default value is 31124.

Upgrading agents and domain manager using a silent installation
Use a silent installation if you want to upgrade agents and satisfy the following objectives:

Perform the upgrade in a safe way
It checks all the processes that are running before starting. It does not perform the upgrade if there are command lines currently running and advises you if there are jobs running. In this case you can decide to wait before performing the upgrade or quit the upgrade. See "Performing a safe upgrade" on page 121 for detailed information.

Upgrade the master, the backup master, and the agents using the same installation method
It can be run on all the workstations of your network.

Use a method that runs unattended and in background
It uses a response file that you customize by adding all the configuration settings to be used during installation. Then, from a command line, running the setup command. Using this method the you can run the installation unattended and in the background.

Manage both UNIX and Windows operating system workstations
It runs both on UNIX and Windows agents.

To upgrade an agent from version 8.3 and higher using a silent installation, follow the procedure described in "Performing a silent installation" on page 84 with the appropriate response files:
Agent
TWS86_UPGRADE_Agent_WIN.txt
TWS86_UPGRADE_Agent_UNIX.txt

Agent with connector
TWS86_UPGRADE_Connector_and_FTA_WIN.txt
TWS86_UPGRADE_Connector_and_FTA_UNIX.txt

Note: The number of minutes that the product waits for jobs that are running to complete before starting the upgrade is 60. If the jobs do not complete during this interval the upgrade does not proceed and an error message is displayed. If you want to change this value customize the `-w checkJobsLoop.wait=60` parameter as indicated in the response file. For details about response file properties, see "The Tivoli Workload Scheduler response file properties" on page 292.

Upgrading agents and domain manager using twsinst
Use twsinst if you want to upgrade agents and satisfy the following objectives:

Perform the upgrade in a safe way
It checks all the processes that are running before starting. It does not perform the upgrade if there are command lines currently running and advises you if there are jobs running. In this case you can decide to wait before performing the upgrade or quit the upgrade. See "Performing a safe upgrade" on page 121 for detailed information.

Save time, disk space, and RAM when upgrading the product
It performs the agent upgrade in 30% less time than the upgrade wizard. It saves disk space and RAM because it is not Java-based.

Use a very simple command
It consists of a single line command.

Manage both UNIX and Windows operating system workstations
It runs both on UNIX and Windows agents.

Use twsinst to upgrade the Tivoli Workload Scheduler agent in your distributed or end-to-end network and add dynamic scheduling capabilities or the Java runtime to run job types with advanced options to the agent. The runtime environment:
- Runs on the agent job types with advanced options, both those supplied with the product and the additional types implemented through the custom plug-ins.
- Enables the capability to remotely run, from the agent, the dynamic workload broker resource command on the server.

See "Software packages and parameters" on page 97 for a description of the parameters. To add dynamic scheduling capabilities, specify the `-tdwbport` and `-tdwghostname` parameters as described in the corresponding description. To add the Java runtime to run job types with advanced options to the agent, specify the `-addjruntime` parameter as described in the corresponding description.

For information about agents installed using twsinst, see "Installing agents using twsinst" on page 88. See http://www.ibm.com/support/docview.wss?rs=672&uid=swg27012175 for a list of supported operating systems and requirements.

Note: After you use the twsinst script to upgrade an instance of Tivoli Workload Scheduler to version 8.6, whose previous version was installed using the installation wizard, you can only use the twsinst method to upgrade that instance of Tivoli Workload Scheduler again. However, you can uninstall that Tivoli Workload Scheduler instance using the wizard.
Running the upgrade:
During the upgrade process, `twsinst` creates a file in the following directories for each of the installation steps:

**On UNIX and Linux operating systems**
```
/user/s_home/TWS
```

**On Windows operating systems**
```
C:\%Program Files%\IBM\TWA
```

If you stop and restart the installation, the installation process starts from the installation step where it was stopped.

To upgrade agents using the `twsinst` script, perform the following steps:

**On UNIX and Linux operating systems**
1. Insert the installation DVD according to the operating system. See “Installation media” on page 26
2. From the `DVD_root/TWS/operating_system` directory, run the `twsinst` script using the synopsis described below.

**On Windows operating systems**
1. Insert the DVD for your operating system. See “Installation media” on page 26.
2. Log in as administrator on the workstation where you want to upgrade the product.
3. From the `DVD_root/TWS/operating_system` directory of the DVD, run `twsinst` using the synopsis described below.

Note: `twsinst` for Windows is a Visual Basic Script (VBS) that you can run in CScript and WScript mode.

A successful upgrade using the `twsinst` issues the return code RC = 0. A failed upgrade issues the return code RC = 1. In the case of a failed installation, see the Tivoli Workload Automation: Messages and Codes.

**Synopsis:**

**On UNIX and Linux operating systems**

Show command usage and version
```
./twsinst -u | -v
```

Upgrade an instance
```
./twsinst -update -uname user_name
[-addruntime boolean]
[-backup_dir backup_dir]
[-displayname agentname]
[-hostname host_name]
[-inst_dir install_dir]
[-jmport port_number]
[-jmportsssl boolean]
[-lang lang-id]
[-nobackup]
[-reset_perm]
[-skip_usercheck]
[-tdwbhostname host_name]
[-tdwbport port_number]
[-wait minutes]
```

**On Windows operating systems:**
Show command usage and version

twinst -u | -v

Upgrade an instance

twinst -update -uname user_name
   -password user_password
   [-addjruntime boolean]
   [-backup_dir backup_dir]
   [-displayname agentname]
   [-domain user_domain]
   [-hostname host_name]
   [-inst_dir install_dir]
   [-import port_number]
   [-importssl boolean]
   [-lang lang_id]
   [-nobackup]
   [-skip_usercheck]
   [-tdwhostname host_name]
   [-tdwpport port_number]
   [-wait minutes]

-\texttt{addjruntime}

Adds the Java runtime to run job types with advanced options to the agent.
The runtime 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. See \textit{Software packages and
parameters} on page 97 for a description of the parameters. Valid values are
true and false. The default is \texttt{false}.

-\texttt{backup_dir} backup_dir

The directory, which must be created manually, where to store the backup copy
of a previous version. If the upgrade fails, you cannot restore your previous
version using the files stored here; you must call IBM Software Support and
provide this path.

If you do not specify this option when running an upgrade, the following
default value is used:

$\texttt{BACKUP\_DIR} = $\texttt{INST\_DIR\_backup}\_\texttt{TWS\_USER}$

where:

- $\texttt{INST\_DIR}$ is the installation path (the user home directory on UNIX and
  Linux).
- $\texttt{TWS\_USER}$ is the user name.

For example:

$\texttt{INST\_DIR}=/opt/TWS/TWS84$
$\texttt{TWS\_USER}=user84$
$\texttt{BACKUP\_DIR}=/opt/TWS/TWS83\_backup\_user83$
$\texttt{BACKUP\_SUBDIR}=/opt/TWS/TWS83\_backup\_user83/TWS84$

-\texttt{displayname}

The name to assign to the dynamic agent. The default is the host name of this
computer.

-\texttt{domain user\_domain}

Windows only. The domain name of the Tivoli Workload Scheduler user. The
default is the name of the workstation on which you are upgrading the
product.

-\texttt{hostname}

The fully qualified hostname on which the agent is contacted by the dynamic
workload broker.
-inst_dir install_dir
The directory where you installed Tivoli Workload Scheduler. When upgrading
inst_dir is used:
• If the upgrade process cannot retrieve the product install location from the
  registries.
• If you need to create the Tivoli Workload Scheduler registries again before
  upgrading. See “Re-creating registry files using twsinst” on page 157 for
details.

If you do not provide the inst_dir and Tivoli Workload Scheduler cannot
retrieve it from the installation registries, the product is installed in the user
home directory.

On UNIX and Linux operating systems:
The path cannot contain blanks. If not specified, the path is set to the
user_name home directory,

On Windows operating systems:
If you specify a path that contains blanks, enclose it in double quotes.
If not specified, the path is set to %ProgramFiles%\IBM\TWA.

-jmport
The port used by the Tivoli Workload Scheduler for z/OS server or the
dynamic workload broker to connect to the Tivoli Workload Scheduler agent.
The default value is 31114. The valid range is from 1 to 65535.

-jmportssl
The port used by the Tivoli Workload Scheduler for z/OS controller, or by the
dynamic workload broker to connect to the Tivoli Workload Scheduler agent.
This number is registered in the ita.ini file located in ITA\cpa\ita on
Windows andITA/cpa/ita on UNIX. For communication using SSL, set jmportssl = 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 jmport
communicates in HTTPS. If you specify true, ensure that you also configure
the HTTPS communication on the z/OS controller. Specify false for HTTP
communication. In this case the port specified in jmport communicates in
HTTP. The default value is true. For communication without using SSL, set
jmportssl = false. To increase the performance of the Tivoli Workload Scheduler
for z/OS server, it is recommended that you set this value to false.

-lang
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.

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 twsinst script.

-nobackup
The upgrade process does not back up the instance you are upgrading.

-password user_password
Windows only. The password of the user for which you are upgrading Tivoli
Workload Scheduler.

-reset_perm
UNIX only. Reset the permissions of the libatrc library.
-skip_usercheck
Enable this option if the authentication process within your organization is not standard, thereby disabling the default authentication option. On UNIX, skip the check of the user in the /etc/password file or using the su command. On Windows, 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.

-tdwbhostname
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 Tivoli Workload Scheduler. If not specified you cannot run your workload dynamically and this parameter assumes the localhost default value. This value is registered in the ResourceAdvisorUrl property in the JobManager.ini file.

-tdwbport
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. The default value is 0, however, if you leave the value as 0, you cannot run your workload dynamically. Specify a nonzero value to add dynamic capability. The valid range is 0 to 65535.

-uname username
The name of the user for which Tivoli Workload Scheduler 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 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.

Examples:
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 runtime to run job types with advanced options:
./twsinst -update -uname twsuser

To upgrade a version 8.5 agent installed in the path /opt/IBM/TWA and give it dynamic scheduling capabilities, but not the Java runtime to run job types with advanced options:

On UNIX and Linux operating systems:
./twsinst -update -uname twsuser -tdwbhostname mybroker.mycompany.com -tdwbport 31116 -inst_dir /opt/IBM/TWA

On Windows operating system:
twsinst -update -uname TWS_user -password qaz12qaz -tdwbhostname mybroker.mycompany.com -tdwbport 31116 -inst_dir "c:\Program Files\IBM\TWA"
To upgrade a version 8.5 agent and give it dynamic scheduling capabilities, and
the Java runtime to run job types with advanced options. The runtime 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 UNIX and Linux operating systems:**

```
./twsinst -update -uname twsuser -tdbhost mybroker.mycompany.com
              -tdbport 31116 -addjruntime true
```

**On Windows operating system:**

```
twsinst -update -uname TWS_user
              -password qaz12qaz
              -tdbhost mybroker.mycompany.com -tdbport 31116 -addjruntime true
              -inst_dir "c:\Program Files\IBM\TWA"
```

### Upgrading agents using Software Distribution

This section describes how to upgrade Tivoli Workload Scheduler agents using
Software Distribution software package blocks. The upgrade process using
Software Distribution software package blocks does not verify if there are
command lines, processes or job running before starting the upgrade you have to
stop them manually by following the procedure described in "Unlinking and
stopping Tivoli Workload Scheduler when upgrading agent workstations" on page
142.

During the upgrade, you can add the following capabilities:

- Standard agent, fault-tolerant agent, or domain manager capabilities
- Dynamic scheduling capabilities
- The option to add the Java runtime to run job types with advanced options to
  the agent. The runtime environment is used to run job types with advanced
  options on the agent and to enable the capability to remotely run, from the
  agent, the dynamic workload broker resource command on the server.

### Creating and installing the software package block:

To create, import, and install the software package block (SPB), complete the
following steps:

1. Create a software package profile that has the following name:
   ```
   FP_TWS_operating_system_<TWS_user>.8.6.00
   ```
   where: `operating_system` is the operating system where you are installing and
   `<TWS_user>` is the user of the installation.

   When you import the software package block, you must pass the name of the
   profile to `wimpspo` so that the Configuration Manager endpoint catalogs the
   name correctly.

2. Import the software package block using the `wimpspo` command.

3. Install the software package block using the `wdinstsp` command.

   **Note:** When upgrading using the `wdinstsp` command, make sure that you
   specify the `install_dir` variable. If you installed the previous version in a
   directory other than the default and you do not specify `install_dir`, Tivoli
   Workload Scheduler is installed as a fresh installation.

For complete instructions about performing these tasks, see the IBM Tivoli
Configuration Manager, Reference Manual for Software Distribution and the IBM Tivoli

### Upgrading procedure overview:
To upgrade, after installing the software package block, perform the following procedure:

1. Upgrade the Common Inventory Technology (CIT). See "Upgrading the Common Inventory Technology."

2. Upgrade the dynamic capabilities of your agent if you installed the version 8.5.1 fault-tolerant agent with dynamic capabilities. See "Upgrading the Tivoli Workload Scheduler dynamic agent" on page 153.

3. Upgrade the fault-tolerant agent. Perform this step to include standard agent, fault-tolerant agent, or domain manager capabilities. See "Upgrading the standard agent, fault-tolerant agent, or domain manager" on page 153.

4. Optional. If you want to add the Java runtime to run job types with advanced options to the agent, see "Upgrading the Java runtime to run job types with advanced options" on page 155.

Some Tivoli Workload Scheduler parameters are used by the software package block to perform the upgrade. You can assign values to each variable to reflect the installation that is being upgraded, otherwise the default value is assigned.

When you upgrade agents using Software Distribution, the following variables are required:

- `install_dir`
- `tws_user`
- `pwd` (This parameter is not required in a UNIX upgrade.)
- `fresh_install`
- `upgrade`
- `from_release`

For a list of Software Distribution parameters, see “Software packages and parameters” on page 97.

**Upgrading the Common Inventory Technology:**
You must upgrade the Common Inventory Technology (CIT) before upgrading the agent and adding the following capabilities:

- Standard agent, fault-tolerant agent, or domain manager capabilities
- Dynamic scheduling capabilities
- The option to add the Java runtime to run job types with advanced options to the agent. The runtime environment is used to:
  - Add the Java runtime to run 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.

The following are examples of the commands that you run to install CIT on Windows and UNIX workstations. See “Software packages and parameters” on page 97 for a description of the parameters.

**Windows operating systems:**
1. `wdinstsp -D CIT_ExploiterID=TWA D:\TWS_86\WINDOWS\CIT_Preinstall.spb`
2. `wdinstsp D:\TWS_86\WINDOWS\CIT.spb`

**UNIX and Linux operating systems:**
1. `wdinstsp -D CIT_ExploiterID=TWA /TWS_86/UNIX/CIT_Preinstall.spb`
2. `wdinstsp /TWS_86/UNIX/CIT.spb`
Upgrading the Tivoli Workload Scheduler dynamic agent:
The Tivoli Workload Scheduler dynamic agent has replaced the fault-tolerant agent with dynamic capabilities available in V8.5.1. To upgrade the latter with `twsinst` you need to install the new dynamic agent.

1. Verify the authorizations required to run the procedure in the "User authorization requirements" on page 59 section.
2. Locate the .spb as described in the "Software packages and parameters" on page 97 section.
3. Set the Software Distribution environment launching the following command located in the TWA/TWS/_uninstall/CLI directory:

   **Windows operating systems:**
   ```bash
   swd_env.bat
   ```

   **UNIX and Linux operating systems:**
   ```bash
   swd_env.sh
   ```

4. Run the `wdinstsp` command located in the TWA/TWS/_uninstall/CLI as shown in the following examples:

   **Windows operating systems:**
   The following Windows example describes an installation with the user `<TWS_user>` and the endpoint `Tivoli_TWS_WINDOWS`. In this example, you are installing on a domain controller or in a Windows node agent because the `-D domain="domain_name"` was specified.

   ```bash
   wdinstsp
   -D ita_port="31112"
   -D host_name=IT041924-T61.rot.ibm.com
   -f
   -uy
   -D install_dir="C:\ibm\TWS\twuser\TWS"
   -D tws_user="twuser"
   -D password="twspasswd"
   -D startAgent="true"
   -D company="company_name"
   -D this_cpu="cpu_name"
   -D master_cpu="MTMDM"
   -D tcp_port="33311"
   -D domain="domain name"
   -n "FP_TWS_LWA_WINDOWS_twuser.8.6.0.00"
   "C:\Output\TWS_VLAST\WINDOWS\Tivoli_LWA_WINDOWS.SPB"
   ```

   **UNIX and Linux operating systems:**
   The following UNIX example describes an installation with the user `<TWS_user>` and the endpoint `Tivoli_TWS_LINUX_I386`.

   ```bash
   wdinstsp
   -D ita_port="31112"
   -D host_name=IT041924-T61.rot.ibm.com
   -f
   -uy
   -D install_dir="/home/twuser/TWS"
   -D tws_user="twuser"
   -D company="company_name"
   -D this_cpu="cpu_name"
   -D master_cpu="MTMDM"
   -D tcp_port="33311"
   -D serverName="server1"
   -n "FP_TWS_LWA_LINUX_I386_twuser.8.6.0.00"
   /mnt/gsa/home/s/l/user1/web/public/SPB_INSTALL/LINUX_I386/Tivoli_LWA_LINUX_I386.SPB"
   ```

Upgrading the standard agent, fault-tolerant agent, or domain manager:
The following are examples of the settings required to upgrade a Tivoli Workload Scheduler agent on Windows and UNIX workstations giving it standard agent, fault-tolerant agent, or domain manager capabilities. You can also add the Java runtime to run job types with advanced options to the agent. The runtime 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. See "Upgrading the Java runtime to run job types with advanced options" on page 155 for details.

1. Verify the authorizations required to run the procedure in the "User authorization requirements" on page 59 section.
2. Locate the .spb as described in the "Software packages and parameters" on page 97 section.
3. Set the Software Distribution environment launching the following command located in the TWA/TWS/_uninstall/CLI directory:

   **Windows operating systems:**
   ```
   swd_env.bat
   ```

   **UNIX and Linux operating systems:**
   ```
   swd_env.sh
   ```

4. Run the `wdinstsp` command located in the TWA/TWS/_uninstall/CLI as shown in the following examples:

   **Windows operating systems:**
   In this example, you are upgrading on a domain controller or in a Windows node agent because the `-D domain="domain_name"` is specified. The following Windows example describes an upgrade from version 8.4 with the user `<TWS_user>` and the endpoint `Tivoli_TWS_WINDOWS`.

   ```
   wdinstsp
   -n "FP_TWS_WINDOWS_twsuser.8.6.0.00"
   -D install_dir="C:\ibm\TWS\twsuser\TWS"
   -D tws_user="twsuser"
   -D password="twspasswd"
   -f
   -uy
   -D company="company_name"
   -D this_cpu="IT041924-T61"
   -D master_cpu="MTMDM"
   -D fresh_install="false"
   -D upgrade="true"
   -D tcp_port="33311"
   -D domain="domain_name"
   -D from_release="8.4"
   "C:\Output\TWS_VLAST\WINDOWS\Tivoli_TWS_WINDOWS.SPB"
   ```

   **UNIX and Linux operating systems**
   The following UNIX example describes an upgrade from version 8.4 with the user `<TWS_user>` and the endpoint `Tivoli_TWS_LINUX_I386`.

   ```
   wdinstsp
   -n FP_TWS_WINDOWS_twsuser.8.6.0.00
   -f
   -uy
   -D install_dir="/home/twsuser/TWS"
   -D tws_user="twsuser"
   -D company="company_name"
   -D this_cpu="IT041924-T61"
   -D master_cpu="MTMDM"
   -D fresh_install=false
   -D upgrade=true
   -D tcp_port="33311"
   ```
Upgrading the Java runtime to run job types with advanced options:
The following are examples of the settings required to upgrade the Java runtime to run job types with advanced options to the agent if you installed it with the version 8.5.1 of the fault-tolerant agent with dynamic capabilities. The runtime environment:

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

See “Software packages and parameters” on page 97 for a description of the parameters.

1. Verify the authorizations required to run the procedure in the “User authorization requirements” on page 59 section.
2. Locate the .spb as described in the “Software packages and parameters” on page 97 section.
3. Set the Software Distribution environment launching the following command located in the TWA/TWS/_uninstall/CLI directory:

   **Windows operating systems:**
   
   swd_env.bat

   **UNIX and Linux operating systems:**
   
   swd_env.sh

4. Run the wdinstsp command located in the TWA/TWS/_uninstall/CLI as shown in the following examples:

   **Windows operating systems:**
   
   The following Windows example describes an upgrade with the user `<TWS_user>`:
   
   wdinstsp
   
   -n "TWS_Eclipse_twsuser.8.6.0.00"
   -D tws_user="twsuser"
   -D from_release="8.4"
   -D install_dir="D:\IBM\TWA\TWS"
   D:\output\TWS_851\WINDOWS\Tivoli_Eclipse_WINDOWS.SPB

   **UNIX and Linux operating systems:**
   
   The following UNIX example describes an upgrade with the user `<TWS_user>`:
   
   wdinstsp
   
   -n "TWS_Eclipse_twsuser.8.6.0.00"
   -D tws_user="twsuser"
   -D from_release="8.4"
   -D install_dir="D:\IBM\TWA\TWS"
   /mnt/sga/home/s/l/user1/web/Tivoli_Eclipse_LINUX_I386.SPB

**Upgrading a command line client**

This section describes how to upgrade a command line client.

To upgrade a Tivoli Workload Scheduler version 8.3 and later command line client using the installation wizard, run the setup for the operating system on which you are upgrading:
On Windows operating systems:
  TWS\operating_system\SETUP.exe or TWS\SETUP.cmd

On UNIX and Linux operating systems:
  TWS/operating_system/SETUP.bin or TWS/SETUP.sh

Alternatively, start the launchpad as follows and select the Tivoli Workload Scheduler installation:

Windows operating system
  From the root directory of the DVD, run launchpad.exe.

UNIX and Linux operating systems
  From the root directory of the DVD, run launchpad.sh.

If you have autorun enabled, the launchpad starts automatically. If you want to start the launchpad from a mounted file system, ensure that you have write permission on it before starting the launchpad.

When the installation wizard is launched, follow the prompts to complete the upgrade.

To upgrade a command line client using the silent installation, follow the procedure described in "Performing a silent installation" on page 84 using the TWS86_UPGRADE_CLI.txt response file.

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 Tivoli Workload Scheduler cannot be found, this can caused by a corrupt registry file. It is possible to upgrade a stand-alone, fault-tolerant agent that has a corrupt registry files without having to reinstall the product. Tivoli Workload Scheduler has a recovery option you can run to recreate 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 when using any of the following upgrade methods:
  • Upgrading using the installation wizard
  • Performing a silent upgrade of the agent
  • Upgrading the agent using the twsinst script

Re-creating the registry files using the installation wizard

If you want to re-create the registry files while performing an upgrade using the installation wizard, perform the following steps:
1. Insert the installation DVD for your operating system.
2. Run one of the following commands:

Windows operating system
  Choose one of the following:
    • TWS\operating_system\SETUP.exe -W InstallationActions.TWA_INSTANCE_PATH=TWA_home -W recovInstReg.run=true
- `TWS\SETUP.cmd -W InstallationActions.TWA_INSTANCE_PATH=TWA_home
  -W recovInstReg.run=true`

  where `operating_system` is the operating system where you want to upgrade Tivoli Workload Scheduler and `TWA_home` is the directory where Tivoli Workload Scheduler is installed.

**UNIX and Linux**

Choose one of the following:

- `TWS\operating_system\SETUP.bin -W InstallationActions.TWA_INSTANCE_PATH=TWA_home -W recovInstReg.run=true`
- `TWS/SETUP.sh -W InstallationActions.TWA_INSTANCE_PATH=TWA_home -W recovInstReg.run=true`

  where `operating_system` is the operating system where you want to upgrade Tivoli Workload Scheduler and `TWA_home` is the directory where Tivoli Workload Scheduler is installed.

Re-creating registry files using the silent upgrade

To re-create the registry files while upgrading an agent using the silent upgrade method, perform the following steps:

1. Copy one of the following response files to a local directory, according to your operating system:
   - `TWS86_UPGRADE_Agent_UNIX.txt`
   - `TWS86_UPGRADE_Agent_WIN.txt`
2. Edit the response file by specifying the parameter: `-W recovInstReg.run=true`
3. Save the file with your changes.
4. Enter the following command:

   **Windows operating system**
   ```
   SETUP.exe -options local_dir\response_file.txt -silent
   ```

   where `response_file.txt` is the name of the response file to use for the upgrade. The `SETUP.exe` file is located in `TWS\operating_system`.

   **UNIX and Linux operating systems**
   ```
   ./SETUP.bin -options local_dir/response_file.txt -silent
   ```

   where `response_file.txt` is the name of the response file to use for the upgrade. The `SETUP.bin` file is located in `TWS/operating_system`.
5. Review the installation messages in the `summary.log` file to check that upgrade was successful.

Re-creating registry files using `twsinst`

To re-create the registry files while upgrading an agent using the `twsinst` script, perform the following steps:

**On UNIX and Linux**

1. Insert the installation DVD according to the operating system.
2. From the `TWS/operating_system` directory, where `operating_system` is the operating system where you want to upgrade Tivoli Workload Scheduler, run the `twsinst` script using the synopsis described below.

**On Windows operating system**

1. Insert the DVD for your operating system.
2. Log in as administrator on the workstation where you want to upgrade the product.

3. From the TWS/operating_system directory of the DVD, where operating_system is the operating system where you want to upgrade Tivoli Workload Scheduler, run twsinst using the synopsis described below.

Synopsis:

On UNIX and Linux

Show command usage and version
    ./twsinst -u | -v

Upgrade an instance
    ./twsinst -update -uname user_name
        [-inst_dir install_dir]
        [-recovInstReg true]]

Example
    ./twsinst -update -uname twsuser -inst_dir /opt/IBM/TWA
        -recovInstReg true

On Windows operating systems:

Show command usage and version
    twsinst -u | -v

Upgrade an instance
    twsinst -update -uname user_name
        -password user_password
        [-domain user_domain]
        [-recovInstReg true]
        [-inst_dir install_dir]

Example
    twsinst -update -uname twsuser -password qaz12qaz
        -inst_dir "C:\Program Files\IBM\TWA" -recovInstReg true

For information about the twsinst parameters, see "Running the upgrade" on page 147.

Adding a feature

To some of the components of the product you have upgraded you can add a feature:

Add a connector to an agent
    If you want to access the plan which is available on an agent or domain manager workstation, add the connector feature which enables the communication between it and the Dynamic Workload Console. See “Adding a connector” on page 166.

Add a language pack to the command-line client
    If you want to use the command-line client in a supported language other than English or the language of the locale of the computer where you installed it, add the language pack. See “Adding language packs to a command line client” on page 167.

Add the Java runtime to an agent
    During the installation of the agent you might have chosen not to add the Java runtime that supports the running of job types with advanced options. If you later decide that you require this function, you can just add the Java runtime separately. See “Adding the Java runtime” on page 102.
If you already installed your environment and want to enable dynamic scheduling capabilities, see “Enabling dynamic scheduling after installation” on page 168.

Configuring

This chapter describes configuring after the installation is complete. It is divided into the following sections:

- “Setting the environment variables” on page 160
- “Configuring a master domain manager” on page 161
- “Configuring a backup master domain manager” on page 161
- “Configuring a domain manager” on page 161
- “Configuring a backup domain manager” on page 162
- “Configuring a dynamic domain manager” on page 163
- “Configuring a backup dynamic domain manager” on page 163
- “Configuring a fault-tolerant agent” on page 164
- “Configuring a dynamic agent” on page 165
- “Configuring a command line client” on page 165
- “Configuring WebSphere Application Server” on page 165
- “Adding a feature” on page 165
- “Enabling dynamic scheduling after installation” on page 168

Setting the environment variables

Before you configure your Tivoli Workload Scheduler components, you must set the environment variables.

On Windows operating systems, run the `tws_env.cmd` shell script to set up both the `PATH` and `TWS_TISDIR` variables. For example, if Tivoli Workload Scheduler is installed in the `%ProgramFiles%\IBM\TWA\TWS` directory, the `PATH` variable is set as follows:

```
c:\Program Files\IBM\TWA\TWS;c:\Program Files\IBM\TWA\TWS\bin
```

**Note:** If you have more than one version of Tivoli Workload Scheduler 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 `tws_env` shell script to set up both the `PATH` and `TWS_TISDIR` variables. For example, if Tivoli Workload Scheduler is installed in the default directory `/opt/IBM/TWA/TWS` directory, `tws_env.sh` sets the variables as follows:

```
PATH=/opt/IBM/TWA/TWS:/opt/IBM/TWA/TWS/bin:$PATH
export PATH
TWS_TISDIR=/opt/IBM/TWA/TWS
    export TWS_TISDIR
```

The `tws_env` script has two versions:

- `tws_env.sh` for Bourne and Korn shell environments
- `tws_env.csh` for C Shell environments
Configuring a master domain manager

After you have installed a master domain manager, if you did not select to automatically add the final job stream during installation and want to do so, follow the steps in this section to add the final job stream to the database and run JnextPlan. This job stream is placed in production every day and runs JnextPlan prior to the start of a new day. The installation creates the FINAL file in the /TWA/TWS directory on your workstation containing the final job stream definition. You can use FINAL or create and customize a new file. See Tivoli Workload Scheduler: User’s Guide and Reference for details about customizing the final job stream.

The following is an example of how to configure a master domain manager after installation:

1. Log in as <TWS_user>.
2. Set the environment variables. See “Setting the environment variables” on page 159.
3. Run the composer command.
4. Add the final job stream definition to the database by running the following command:
   ```
   add FINAL
   ```
   where FINAL is the name of the file containing the definition of the Final job stream.
5. Exit the composer command line.
6. Run the JnextPlan job:
   ```
   JnextPlan
   ```
   You can automate this step after installation. See Tivoli Workload Scheduler: User’s Guide and Reference.
7. When JnextPlan completes, check the status of Tivoli Workload Scheduler:
   ```
   conman status
   ```
   If Tivoli Workload Scheduler started correctly the status returned by the command is Batchman=LIVES.
8. Raise the workstation limit value to allow jobs to run. The default job limit after installation is 0, so no jobs are permitted to run at a time. Raise the job limit to allow jobs to run, for example to 10 jobs:
   ```
   conman "limit ;10"
   ```
   If no workstation name is specified for the limit command, the default value is the current login workstation.

**Note:** If priority of jobs is equal to HI (100) or GO (101), they will disregard the limit and run despite a limit=0, unless fence>=priority.

Additionally, the following configuration procedures might be necessary. For information about these procedures, see Tivoli Workload Scheduler: 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 or GSKit for inter-component communications.
Configuring a backup master domain manager

After you have installed a backup master domain manager, perform the following configuration steps:
1. Log in as <TWS_user> on your master domain manager.
3. Set the environment variables by running tws_env.
4. Define the backup master as a full status autolink fault-tolerant agent in the Tivoli Workload Scheduler database, using the composer command interface or the Dynamic Workload Console. In this example using composer:

```bash
composer
new
```
5. Type the workstation definition in the text editor, for example:

```bash
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 Tivoli Workload Scheduler: User’s Guide and Reference for information.
6. Run JnextPlan -for 0000 to include the backup master workstation in the plan and to send the Symphony file to it.

**Note:** Ensure that the global option carryforward is set to all or only the unfinished jobstreams will be 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:

```bash
conman "limit DM1;10"
```

**Note:** If you are logged into the backup master, DM1 is not required.

Additionally, the following configuration procedures might be necessary. For information about these procedures, see Tivoli Workload Scheduler: 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 or GSKit for inter-component communications.

Configuring a domain manager

After you have installed 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 tws_env.
3. Define the domain manager as a full status autolink fault-tolerant agent in the Tivoli Workload Scheduler 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 DDM1
  DESCRIPTION "domain manager"
  OS UNIX
  NODE lab0777
  TCPADDR 31111
  DOMAIN MDM
  FOR MAESTRO
    TYPE MANAGER
    AUTOLINK ON
    BEHINDFIREWALL OFF
    FULLSTATUS ON
END
```

For more information about workstation definitions, see the Tivoli Workload Scheduler: User’s Guide and Reference for information.

5. 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` or only the unfinished jobstreams will be 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"
```

### Configuring a backup domain manager

After you have installed 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 `tws_env`.
3. Define the backup domain manager as a full status autolink fault-tolerant agent in the Tivoli Workload Scheduler 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 DDM1
  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 Tivoli Workload Scheduler: User’s Guide and Reference for information.
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` or only the
   unfinished jobstreams will be 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"

### Configuring a dynamic domain manager

After you have installed 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 `tws_env`.
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` or only the
   unfinished jobstreams will be 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"

### Configuring a backup dynamic domain manager

After you have installed a backup 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 `tws_env`.
3. Define the backup dynamic domain manager as a full status autolink
   fault-tolerant agent in the Tivoli Workload Scheduler 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 *Tivoli Workload

5. Run `jnextplan -for 0000` to include the backup 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 or only the unfinished jobstreams will be 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:
   `composer "limit F235007_00;10"

Configuring a fault-tolerant agent

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. See the Tivoli Workload Scheduler: User’s Guide and Reference for information. 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 `tws_env.sh`.
3. Create the workstation definition in the Tivoli Workload Scheduler 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 Tivoli Workload Scheduler Reference Guide.

Note: Ensure that the global option carryforward is set to all or only the unfinished jobstreams will be carried forward.

5. 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"

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:
   `composer "limit F235007_00;10"

Additionally, the following configuration procedures might be necessary. For information about these procedures, see Tivoli Workload Scheduler: 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 or GSKit for inter-component communications.

Configuring a dynamic agent
After installing a dynamic agent, perform the following steps:
1. Run JnextPlan with the option -for 0000 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 User’s Guide and Reference.
   
   Note: Ensure that the global option carryforward is set to all otherwise only the unfinished jobstreams are carried forward.

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:
   
   composer "limit F235007_00;10"

   Additionally, the following configuration procedures might be necessary. For information about these procedures, see Administration Guide.
   • Customizing and configuring global, local, and user options.
   • Customizing and configuring user authentication to allow users authorization for actions and objects, and to configure LDAP.
   • Setting connection security to enable SSL or GSKit for inter-component communications.

Configuring a command line client
The following configuration procedures might be necessary for a command line client. For information about these procedures, see Tivoli Workload Scheduler: Administration Guide.
   • Customizing and configuring global and local options
   • Customizing and configuring user authentication to allow users authorization for actions and objects, and to configure LDAP
   • Setting connection security to enable SSL or GSKit for inter-component communications

Configuring WebSphere Application Server
If, after installing, you have more than one instance of WebSphere Application Server managing any Tivoli Workload Automation products, you must ensure that they have the same LTPA token_keys. See the Tivoli Workload Scheduler: Administration Guide.

Adding a feature
This section explains how to add one of the following features to your environment after you installed or upgraded your network:

Add a connector to an agent
If you want to access the plan which is available on an agent or domain manager workstation, add the connector feature which enables the communication between it and the Dynamic Workload Console. See "Adding a connector" on page 166.

Add a language pack to the command-line client
If you want to use the command-line client in a supported language other
than English or the language of the locale of the computer where you installed it, add the language pack. See "Adding language packs to a command line client" on page 167.

Add the Java runtime to an agent
During the installation of the agent you might have chosen not to add the Java runtime that supports the running of job types with advanced options. If you later decide that you require this function, you can just add the Java runtime separately. See "Adding the Java runtime" on page 102.

Adding a connector
To add a connector instance to an existing installation, perform the following steps:
1. For a graphical installation, from the installation DVD, start the launchpad as described in "Launchpad" on page 30 and select the Tivoli Workload Scheduler installation, or run the setup for the operating system on which you are installing.
   From the DVD, perform the following:
   
   **On Windows operating system:**
   
   TWS\operating_system\SETUP.exe or TWS\SETUP.cmd
   
   **On UNIX and Linux operating systems:**
   
   TWS/SETUP.sh or TWS/operating_system/SETUP.bin
   
   **Note:** SETUP.sh copies the entire image to a temporary directory. Ensure there is enough space available.

2. Follow the installation wizard screens to complete the installation. The following list describes the fields that you might need to complete during the installation.

   **Add a New Feature**
   Select the agent on which you want to add the connector. Add connector is displayed.

   **TWSUser password**
   The password of the <TWS_user>.

   **Note:** If the <TWS_user> of the agent on which you are adding the Connector is different from the WebSphere Application Server administration user you used when you installed the Dynamic Workload Console, you should make a note to pay especial attention when performing administration activities on WebSphere Application Server to always use the WebSphere Application Server administration user's credentials, not the credentials of the <TWS_user>. You should also note that in these circumstances you might experience a small problem during the uninstallation of the Connector (see "The uninstallation of the Connector fails in the "Start the embedded WebSphere Application server" step" on page 232).

   **HTTP transport**
   The port for the HTTP transport. The default value is 31115. The valid range is from 1 to 65535.

   **HTTPS transport**
   The port for the secure HTTPS transport. The default value is 31116. The valid range is from 1 to 65535.
Bootstrap
The port for the bootstrap or RMI. The default value is 31117. The valid range is from 1 to 65535.

SOAP connector
The port for the application server protocol SOAP connector. The default value is 31118. The valid range is from 1 to 65535.

SAS Server Authentication Listener
The port used by the Secure Association Services (SAS) to listen for inbound authentication requests. The default value is 31119. The valid range is from 1 to 65535.

CSIv2 Server Authentication Listener
The port on which the Common Secure Interoperability Version 2 (CSIv2) service listens for inbound server authentication requests. The default value is 31120. The valid range is from 1 to 65535.

CSIv2 Client Authentication Listener
The port on which the Common Secure Interoperability Version 2 (CSIv2) service listens for inbound client authentication requests. The default value is 31121. The valid range is from 1 to 65535.

ORB Listener
The port used for RMI over IIOP communication. The default value is 31122. The valid range is from 1 to 65535.

Administration HTTP transport
The administrative console port. The default value is 31123. The valid range is from 1 to 65535.

Administration HTTPS transport
The administrative console secure port. The default value is 31124. The valid range is from 1 to 65535.

Note: The installation procedure checks for the availability of the ports in the specified port range. If one or more ports are in use by other applications, you are prompted to enter a new port number.

Adding language packs to a command line client
To add language packs to a command line client, perform the following steps:
1. From the installation DVD, run the setup for the operating system on which you are installing:
   - On Windows operating system: TWS\operating_system\SETUP.exe or TWS\SETUP.cmd
   - On UNIX and Linux operating systems: TWS/SETUP.sh or TWS/operating_system/SETUP.bin
2. Select the installation wizard language. Click OK.
3. Read the welcome information. Click Next.
4. Read and accept the license agreement. Click Next.
5. From the drop-down list, select an existing command line client. Existing installations are identified by the installation path. Note that the option Add langPack is automatically selected.
6. Select the additional languages to install. Click Next.
7. Review the installation settings. Click Next.
8. When the installation completes, a panel displays a successful installation or indicates the directory of the log file if the installation was unsuccessful. Click Finish.

**Adding the Java runtime to run job types with advanced options**
Java runtime allows you to run job types with advanced options, both those supplied with the product and the additional types implemented through the custom plug-ins.

If you did not install the Java runtime when you installed the other Tivoli Workload Scheduler components, you can add it at any time by installing the Java Runtime Environment software package block using the `wdinstsp` command. See “Adding the Java runtime” on page 102 for more information.

**Enabling dynamic scheduling after installation**
This section describes the procedure that you must follow to enable dynamic scheduling if upgraded the product, both the master and the agent, without enabling the dynamic scheduling capabilities. For example, you upgraded the product in the following ways:

**Using the installation wizard**
You did not select one or both of the following options:
- **Enable the dynamic scheduling capabilities**, when upgrading the master
- **Dynamic agent**, when upgrading the agent.

**Using twsinst, when upgrading the agent**
You did not specify the `-tdbport dbport_number` and `-tdbhostname host_name`.

To enable dynamic scheduling, perform the following steps:

1. In the `tws_home/TDB/config/BrokerWorkstation.properties` file, modify the values of the following properties according to the values that you specified at upgrade time:

    ```
    Broker.Workstation.Name= master_name_DWB
    Broker.Workstation.Port= port_number
    MasterDomainManager.Name= host_name
    Broker.AuthorizedCNs=server1; ...
    ```

    where:

    **Broker.Workstation.Name=master_name_DWB**
    It is the master domain manager name followed by _DWB. You can modify this value including the _DWB suffix.

    **Broker.Workstation.Port=port_number**
    It is the port on the workload broker workstation used by the Tivoli Workload Scheduler master domain manager to communicate with dynamic workload broker. You can specify any value. The default value is `41114` if the Netman port number is `31111`. The valid range is from 1 to 65535. If you changed the Netman port number, the Broker.Workstation.Port `port_number` is calculated as:
    
    ```
    netman_port_number+10003
    ```

    **MasterDomainManager.Name=host_name**
    It is the fully qualified host name on which the master domain manager will be contacted by the agents.
Broker.AuthorizedCNs=server1; ... ;servern  

It is the list of prefixes of common names master domain manager certificates authorized to communicate with the broker server. For more information about authorizing the connection to the server see the Customizing the SSL connection to the master domain manager and dynamic domain manager section in the Tivoli Workload Scheduler: Administration Guide.

2. On the master domain manager, verify the current value of the httpsPort by running the showHostProperties wastool. The default value is 31116. The following is an example output:

```
#################################################################
# Ports Configuration Panel
#################################################################
bootPort=38317
bootHost=nynewhost.romelab.ibm.it.com
soapPort=38318
soapHost=mynewhost.romelab.ibm.com
httpEnabled=true
httpPort=21115
httpHost=* httpsEnabled=true
httpsPort=31116
```

3. On the master domain manager and on every agent that is connected to the workload broker server, update the JobManager.ini configuration file located under:

- On Windows operating system:  
  `tws_home\TWS\ITA\cpa\config\JobManager.ini`

- On UNIX and Linux operating systems:  
  `tws_home/TWS/ITA/cpa/config/JobManager.ini`

by assigning to the tdwb_hostname and tdwb_httpsport variables contained in the ResourceAdvisorUrl property, the following values:

- `tdwb_hostname`  
  Specify the fully qualified host name of the workload broker server

- `tdwb_httpsport`  
  Specify the value that the httpsPort has on the master domain manager as shown by the showHostProperties wastool. The default is 31116, which is the dynamic workload broker port number. The port is currently set to zero because at installation time you specified that you would not use the dynamic workload broker.

The ResourceAdvisorUrl property has the following syntax:

`ResourceAdvisorUrl = https://<tdwb_hostname>:<tdwb_httpsport>/JobManagerRESTWeb/JobScheduler/resource`

4. Start the dynamic workload broker component by running the startBrokerApplication.sh wastool as follows:

```
/TWS_home/wastools/startBrokerApplication.sh -user user_name -password password
```

where:

- `user_name`  
  Specifies the name of the WebSphere Application Server.

- `password`  
  Specifies the password of the WebSphere Application Server.
5. On the master domain manager and on every agent of your network that you want to connect to the workload broker server, start the Tivoli Workload Scheduler agent by running the following command from the TWS_home directory:

- On Windows operating system:
  ```
  StartUpLwa.cmd
  ```

- On UNIX and Linux operating systems:
  ```
  StartUpLwa
  ```

### Uninstalling

This chapter describes how you uninstall Tivoli Workload Scheduler. It is divided into the following sections:

- “User authorization requirements” on page 59
- “Uninstalling a dynamic domain manager” on page 171
- “Uninstalling using the wizard” on page 171
- “Performing a silent uninstallation” on page 172
- “Uninstalling agents using the twsinst script” on page 173
- “Uninstalling using the Software Distribution CLI” on page 174
- “Uninstalling a command line client” on page 175

The uninstaller program is created during the installation procedure. Wherever possible, use the same method you chose to install the product when uninstalling the product. For example, if you installed the product using the installation wizard, use the uninstaller program to subsequently remove the product.

Uninstalling the product does not remove files created after Tivoli Workload Scheduler 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:**

1. The Tivoli Workload Scheduler engine is a prerequisite for other products and features you can install, such as Tivoli Workload Scheduler for Applications and the connector. Before you uninstall the engine, uninstall all the additional features.

2. See the Tivoli Workload Scheduler: Administration Guide for information about removing Tivoli Workload Scheduler manually.

### User authorization requirements

Check the authorization roles before beginning the procedure.

Authorization requirements for running any of the installation, upgrade, or uninstallation wizards or commands are the same:

**UNIX and Linux operating systems**

- `root` access

**Windows operating system**

- Your login account must be a member of the Windows [Administrators](#) group or domain administrators with [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. To do this, before running the installation, perform the following steps:

1. Right-click the icon that you use to run the program:

   - **If you are using the wizard:**
     
     ```
     SETUP.exe
     ```

   - **If you are using the silent installation or twsinst:**
     
     ```
     Command Prompt
     ```

2. Select Run as administrator.

In addition, to use Software Distribution, the user must have the Software Distribution roles admin, senior, or super.

### Uninstalling a dynamic domain manager

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 chapter.

2. In the database, delete the definition of the workstations of type AGENT that are connected to the dynamic domain manager to be uninstalled. You can use either the Dynamic Workload Console workload designer or run the following command:

   ```
   composer del ws agent_workstation_name
   ```

3. Delete the definition of the workstations of type REM-ENG connected to the dynamic domain manager to be uninstalled. 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 definition of the workstations of type POOL connected to the dynamic domain manager to be uninstalled. You can use either the Dynamic Workload Console workload designer or run the following command:

   ```
   composer del ws pool_workstation_name
   ```

5. Delete the definition of the workstations of type D-POOL connected to the dynamic domain manager to be uninstalled. 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 by using the "Uninstalling using the wizard" or the "Performing a silent uninstallation" on page 172 procedure.

7. Delete the definition of the workstation of type X-AGENT hosted by the dynamic domain manager to be uninstalled. 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 definition of the workstations of type BROKER of the dynamic domain manager to be uninstalled. You can use either the Dynamic Workload Console workload designer or run the following command:

   ```
   composer del ws broker_workstation_name
   ```

### Uninstalling using the wizard

The **uninstaller** program removes product files, registry keys, and services. It also removes the binaries related to the Tivoli Workload Scheduler agent installed, the distributed connector, and the language packs.
To uninstall Tivoli Workload Scheduler, perform the following steps:

1. Ensure that all Tivoli Workload Scheduler processes and services are stopped, and that there are no active or pending jobs. For information about stopping the processes and services see *Administration Guide*.
2. Navigate to the *twshome* path.
3. Run the uninstall script:
   - On Windows operating systems: `uninstaller.exe`
   - On UNIX and Linux operating systems: `./uninstall.bin`
4. Select the Tivoli Workload Scheduler instance you want to uninstall:
   - If you are uninstalling a master domain manager, the wizard removes the selected instance and any additional feature installed for that instance. If you are uninstalling from an integrated Tivoli Workload Automation, the embedded WebSphere Application Server is not removed. Only Tivoli Workload Scheduler applications are removed.
   - If you are uninstalling an agent, you can choose if you want to uninstall the connector only, or both the agent and connector simultaneously.

**Performing a silent uninstallation**

For a silent uninstallation of a master domain manager, a backup master domain manager, a dynamic domain manager, or a backup dynamic domain manager, perform the following steps:

1. Ensure that all Tivoli Workload Scheduler processes and services are stopped, and that there are no active or pending jobs. For information about stopping the processes and services see *Administration Guide*.
2. Navigate to the */TWA/TWS/_uninstall* path.
3. Enter the following command:
   - On Windows operating systems: `uninstaller.exe -silent`
   - On UNIX and Linux operating systems: `./uninstall.bin -silent`

For a silent uninstallation of an agent, a connector, or both, perform the following steps:

1. Ensure that all Tivoli Workload Scheduler processes and services are stopped, and that there are no active or pending jobs.
2. Copy the *TWS86_UNINSTALL_Agent.txt* response file from the installation DVD in the */TWS\RESPONSEFILES* directory to a local directory and edit it as appropriate.
3. Save the file with your changes.
4. Navigate to the */TWA/TWS/_uninstall* path.
5. Enter the following command:
   - On Windows operating systems: `uninstaller.exe -options <local_dir>\TWS86_UNINSTALL_Agent.txt -silent`
   - On UNIX and Linux operating systems: `./uninstall.bin -options <local_dir>/TWS86_UNINSTALL_Agent.txt -silent`
Note: If you want to reinstall after performing a silent uninstallation, you must first close and reopen the shell to correctly reset the environment variables.

**Uninstalling agents using the twsinst script**

Follow these steps to uninstall Tivoli Workload Scheduler agents using the `twsinst` script. Only agents installed using `twsinst` can be uninstalled using `twsinst`. Depending on the operating system, proceed as follows:

- **On UNIX, Linux, and IBM i operating systems:**
  1. Ensure that all Tivoli Workload Scheduler processes and services are stopped, and that there are no active or pending jobs. For information about stopping the processes and services, see *Administration Guide*.
  2. Log on as root (on IBM i operating systems, log on as QSECOFR) and change your directory to `/installation_dir/TWS`. For example: `/home/user1/TWS` where user1 is the name of Tivoli Workload Scheduler user.
  3. From the `installation_dir/TWS` directory, run the `twsinst` script as follows:

```
twsinst -uninst -uname username [-wait minutes] [-lang lang_id]
```

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 `twsinst` script from the `/installation_dir/TWS` (for example, `/home/user1/TWS`) as follows:

```
./twsinst -uninst -uname user_name -lang language
```

where `language` is the language set during the uninstallation.

- **On Windows operating systems:**
  1. Ensure that all Tivoli Workload Scheduler processes and services are stopped, and that there are no active or pending jobs. For information about stopping the processes and services see *Administration Guide*.
  2. Log on as administrator on the workstation where you want to uninstall the product.
  3. From the `installation_dir/TWS` (for example, c:\Program Files\IBM\TWA), run the `twsinst` script as follows:

```
twsinst -uninst -uname username [-wait minutes] [-domain domain_name] [-lang lang_id]
```

**Note:** `twsinst` for Windows is a Visual Basic Script (VBS) that you can run in CScript and WScript mode. 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 `twsinst` script from the `/installation_dir/TWS` (for example, `/home/user1/TWS`) as follows:

```
twsinst -uninst -uname user_name -lang language
```

where `language` is the language set during the uninstallation.

- **uninst**
  Uninstalls Tivoli Workload Scheduler.

- **uname username**
  The name of the user for which Tivoli Workload Scheduler is uninstalled. This
user name is not to same as the user performing the installation logged on as root on UNIX and Linux, as administrator on Windows, and as QSECOFR on IBM i.

-**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.

-**domain domain_name**
Windows only. The domain name of the Tivoli Workload Scheduler user. The default is the name of the workstation on which you are uninstalling the product.

-**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.

The following example shows a twsinst script that uninstalls the Tivoli Workload Scheduler agent installed for user named twsuser:

**On UNIX, Linux, and IBM i operating systems:**

```
./twsinst -uninst -uname TWS_user
```

**On Windows operating systems:**

```
twsinst -uninst -uname TWS_user
```

**Uninstalling using the Software Distribution CLI**

You can uninstall Tivoli Workload Scheduler using a Software Distribution/Configuration Manager command. To uninstall a software package from a disconnected target, use the command wdrmvsp. Tivoli Workload Scheduler uses the disconnected catalog.

Ensure that all Tivoli Workload Scheduler processes and services are stopped, and that there are no active or pending jobs. For information about stopping the processes and services see Administration Guide.

For example, to uninstall on UNIX, perform the following:

```conman
"stop;wait"
conman "shut;wait"
```

Ensure all processes are down.

As root:
```
cd <twshome>/_uninstall/CLI. ./swd_env
```

To display package names and versions: wdlssp

```
wdrmvsp -f packagename.version
```

Using the same procedure, you can also remove the software package block that installs language packs, the Java runtime to run job types with advanced options, the dynamic agent. See “Uninstalling Tivoli Workload Scheduler manually” on page 236 for information about removing Tivoli Workload Scheduler manually.
Uninstalling a command line client

You can uninstall a command line client using the uninstallation wizard or by performing a silent uninstallation. To uninstall a command line client perform the following steps:

1. Navigate to the CLI_home/_uninstall path, where CLI_home is the installation path of your command line client.
2. To uninstall using the wizard, run the uninstaller command:
   - On Windows operating system:
     uninstaller.exe
   - On UNIX and Linux operating systems:
     ./uninstall.bin

Note: For a silent installation, use the -silent flag.

Uninstalling the additional plug-ins using the Tivoli Workload Scheduler for Additional Plug-ins

Uninstalling the additional plug-ins either using the wizard or the silent method.

When you uninstall the additional plug-ins, you can uninstall one or more plug-ins simultaneously.

If you installed an additional plug-in by using the silent installation, uninstall it using the same procedure.

To uninstall an additional plug-in, you can use any of the following procedures:

Wizard
   For details, see “Uninstalling by using the wizard.”

Silent
   For details, see “Uninstalling by using the silent uninstallation” on page 176.

Note: You can uninstall only additional plug-ins installed by using Tivoli Workload Scheduler for Additional Plug-ins.

Uninstalling by using the wizard

Uninstalling the additional plug-ins by using the wizard.

Using the wizard, you can uninstall one or more plug-ins at a time.

If you installed the product using the installation wizard, you can uninstall it either using the uninstallation wizard or the silent uninstallation. If you installed the product using the silent installation you must use the silent uninstallation to uninstall it.

To uninstall one or more plug-ins, perform the following steps:

1. Insert the Tivoli Workload Scheduler for Applications DVD or eImages, for the operating system where you are installing, run the setup installation command:
   - On Windows operating systems:
     From the \PLUGIN_INSTALLER directory,
     setup.bat
   - On UNIX and Linux operating systems:
     From the /PLUGIN_INSTALLER directory,
     ./setup.sh
The uninstall program starts.

2. Select the language for the wizard, and click **OK**. The Welcome panel is displayed.

3. Read the welcome information and click **Next**. The Operations panel is displayed.

4. Select the **Uninstall** radio button. Click **Next**. The Plug-in list panel is displayed.

5. Select the additional plug-ins you want to uninstall and click **Next**. The summary of the plug-ins that you selected for uninstall is displayed.

6. The uninstall process starts. When the uninstallation completes, a panel showing the results, is displayed.

7. Click **Finish** to exit the wizard.

If you received the error messages, analyze the uninstallation log files listed in [Table 13](#).

**Table 13. Uninstallation log files**

<table>
<thead>
<tr>
<th>Log file name</th>
<th>Content</th>
<th>Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>tws4plugins_status.log</td>
<td>The additional plug-in uninstallation status log file is created only for silent uninstallation. It reports if the uninstallation completed successfully or with errors. In case of errors it indicates if the error is due to an incorrect field value or to a failed step.</td>
<td>At the begin of the uninstallation process this log file is created in the following directory:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>On Windows operating systems:</strong> %TEMP%\twa\tws4apps</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>On UNIX and Linux operating systems:</strong> $tmp\twa\tws4apps</td>
</tr>
<tr>
<td></td>
<td></td>
<td>and copied to directory <strong>Tivoli Workload Scheduler_installation_dir\logs</strong> at the end of the uninstallation process.</td>
</tr>
<tr>
<td>tws4plugins_ia_uninstall.log</td>
<td>additional plug-in log file for InstallAnywhere errors.</td>
<td><strong>Tivoli Workload Scheduler_installation_dir\logs</strong></td>
</tr>
<tr>
<td>tws4plugins_uninstall.log</td>
<td>The additional plug-in uninstallation log file.</td>
<td>At the begin of the uninstallation process this log file is created in the following directory:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>On Windows operating systems:</strong> %TEMP%\twa\tws4apps</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>On UNIX and Linux operating systems:</strong> $tmp\twa\tws4apps</td>
</tr>
<tr>
<td></td>
<td></td>
<td>and copied to directory <strong>Tivoli Workload Scheduler_installation_dir\logs</strong> at the end of the uninstallation process.</td>
</tr>
</tbody>
</table>

**Note**: If you are uninstalling in silent mode and you need to see the logs files, check before the `tws4plugins_status.log` file to verify the installation process status and then check the `tws4plugins_install.log` file for details.

**Uninstalling by using the silent uninstallation**

Use the silent uninstallation process to uninstall the additional plug-ins, without the user intervention. Using the silent method you can uninstall all the installed plug-ins simultaneously or one plug-in at a time.
If you installed the plug-in using the installation wizard, you can uninstall it either using the uninstallation wizard or the silent uninstallation. If you installed the plug-ins by using the silent installation you must use the silent method to uninstall it.

When running the uninstallation in silent mode, no messages are displayed. The messages are written in the silent installation log files listed in Table 13 on page 176. If the silent uninstallation fails, you can verify the messages written in the log files.

To uninstall one or more plug-ins at a time, run the following command:

**On Windows operating systems:**
From the /PLUGIN_INSTALLER directory of the DVD or eImages product,
```
setup.bat -i silent -f <response file>
```
Where *response file* is a template file that you can customize to indicate the list of the plug-ins you want to uninstall. The default response file is TWSPlug-ins_ResFile_Uninst_windows.txt. It is located under the RESPONSE_FILE directory.

**On UNIX and Linux operating systems:**
From the /PLUGIN_INSTALLER directory of the DVD or eImages product,
```
./setup.sh -i silent -f <response file>
```
Where *response file* is a template file that you can customize to indicate the list of the plug-ins you want to uninstall. The default response file is TWSPlug-ins_ResFile_Uninst_unix.txt. It is located under the RESPONSE_FILE directory.

Table 14 lists the options you can specify when uninstalling.

<table>
<thead>
<tr>
<th>Option</th>
<th>Required</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
</table>
| USER_INSTALL_DIR="<path>"     | Yes      | Specify the IBM Tivoli Workload Scheduler installation path from where you want to uninstall one or more additional plug-ins. | A fully qualified path. For example, to uninstall the additional plug-ins under C:\Program Files\IBM\TWA86, specify: USER_INSTALL_DIR="C:\Program Files\IBM\TWA86"
**On Windows operating systems:**
The default uninstallation path is "c:\Program Files\IBM\TWA"
**On UNIX and Linux operating systems:**
The default uninstallation path is /opt/IBM/TWA |
### Table 14. Options to perform a silent uninstallation (continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Required</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLUGINS_TO_UNDEPLOY=&lt;plug-in_1&gt;,..., &lt;plug-in_n&gt;</td>
<td>Yes</td>
<td>Specify the plug-ins ID that corresponds to the plug-ins you want to uninstall, separated by comma.</td>
<td>To find the plug-in ID, perform the following actions: 1. Open the <code>&lt;TWA_HOME&gt;\InstallDataPlugins\plugin_&lt;plug-in_name&gt;.xml</code> file 2. Identify the value of the id attribute of the plug-in: <code>&lt;pluginInfo version=&quot;&lt;plug-in_version&gt;&quot; name=&quot;&lt;plug-in_name&gt;&quot; id=&quot;&lt;plug-in_id&gt;&quot; /&gt;</code> For example, to uninstall the following plug-in: <code>&lt;pluginInfo version=&quot;1.0.0.01&quot; name=&quot;plug-in_test_diskspace&quot; id=&quot;plug-in_test_ds&quot; /&gt;</code> specify the value: PLUGINS_TO_UNDEPLOY= plug-in_test_ds</td>
</tr>
<tr>
<td>ACTION_TYPE=&lt;value&gt;</td>
<td>Yes</td>
<td>Specify the action that uninstallation process performs on plug-in. In this case the value must be set to UNDEPLOY.</td>
<td>The value must be set to UNDEPLOY.</td>
</tr>
</tbody>
</table>

The following is an example of the command you run from the directory `/PLUGIN_INSTALLER`, to perform a silent installation on a UNIX workstation, by using the response file `TWSPlug-ins_RespFile_uninst_UNIX.txt`:
```
./setup.sh -i silent -f TWSPlug-ins_RespFile_uninst_UNIX.txt
```

The following example shows a response file that uninstalls the `plug-in_test_ds` on a Windows workstation:
```
USER_INSTALL_DIR=c:\Program Files\IBM\TWA86
FEATURE_UNINSTALL_LIST=plug-in_test_ds
```

### Troubleshooting installation, migration, and uninstallation

This chapter describes issues dealing with the installation, removal, and configuration of Tivoli Workload Scheduler and its prerequisites. It is divided into the following topics:

- "Log files of installation processes” on page 179
- “Recovering a failed interactive InstallShield wizard installation” on page 179
- “Recovering a failed silent InstallShield wizard installation” on page 190
- ‘Recovering a failed upgrade” on page 191
- “Problem scenarios: install, reinstall, upgrade, migrate, and uninstall” on page 193
- “Security implications of the installation” on page 234
- “Verifying the installation” on page 235
Log files of installation processes

Log files of the installation processes have a different naming convention, depending on the installation method. For information about the different log files associated with the different installations, see “Installation log files” on page 248.

Packaging log files for support

If a problem occurs with an installation that you cannot resolve, IBM Software Support might ask you to send them all of the installation log files. Include the following:

- For Tivoli Workload Scheduler, all of the files and subdirectories in the `<tempDir>/TWA/tws86` directory.
- For the Dynamic Workload Console, all of the files and subdirectories in the or the `<tempDir>/TWA/tdwc86` directory.
- The software package block log files.
- The DB2 installation log.
- The installation log of the embedded WebSphere Application Server.
- For Software Developers Kit, all files and subdirectories in the `/tmp/TWA/sdk86` directory.
- For the Job Brokering Definition Console, all files and subdirectories in the `/tmp/TWA/jbdc86` directory.

Note: Do not remove, add, or modify files in the `<tempDir>/TWA/tws86` directory because this might cause an installation to fail, or prevent the recovery of a failed installation.

Recovering a failed interactive InstallShield wizard installation

This section describes how to recover a failed interactive installation.

If an operation fails during the installation, the wizard opens the following panel:
TVT Instructions

To capture this and the following windows in this section, follow this procedure:

1. Chose a computer where DB2 is installed
2. Run the install wizard and select to install a new master domain manager
3. When asked about DB2 select to use the existing instance
4. STOP WHEN YOU GET TO THE SUMMARY PANEL
5. Stop DB2, as follows:
   a. Open a DB2 window
   b. Type "db2 force application all"
   c. Type "db2stop"
   DB2 stops.
6. Click "Next" to let the installation start. When it fails, follow the procedures written in the guide to navigate from screen to screen to make the screen captures - you might need to adjust the size of the screen to get the best results.

**Note:** If you are using the interactive wizard, do not close the wizard panel by clicking on the Close icon: 

You can use the debug mode of the wizard to see which steps of the installation have failed. You can correct errors that have occurred and resume those installation steps that have not completed successfully, without leaving the wizard.

You can choose to do this immediately the failure occurs, or close the window and recover the situation later.

The procedure is described in the following sections:

- "The Step List window" on page 181
- "The Step window" on page 182
- "Correcting a failed step and continuing the installation" on page 186
The Step List window

The Step List window opens either when an installation fails, or when you are resuming an installation that had previously been stopped (see “Stopping and resuming an interactive installation” on page 189). Figure 13 shows an example of the Step List window when an installation step has failed:

![Step List window](image)

**Figure 13. Step List window showing a failed step**

The Step List window is organized as follows:

**Step #** The installation sequence.

**Description** The description of the installation step. The steps of the Tivoli Workload Scheduler installation are described in “Installing” on page 59.

**Target** The workstation where the installation is being run.

**Status** The step status. It can be one of the following:

- **Ready** The step is ready to be installed.
- **Success** The step has successfully completed.
- **Error** The step completed, but errors were detected.
- **Held** A step that prerequisites another step has failed. Do not set this state.

**Run next** Start the next step in the list that has a status set to Ready.
Run all
Start, in sequence, all the steps in the list that have a status set to Ready.

Stop
Use this to stop the step processing while a step is being processed. The step returns to the Ready status.

Stop on error
If selected, stops the processing of any step or steps that you run in the event of an error.

Search by status
Select the status you want to view, then click Search. The step list displays the first step in the step list with the selected status.

Status
The status of the installation processing engine. It can be one of the following:

Waiting
User action is required.

Running
Installation of a step is in progress.

Stopping
After the current step, the engine stops.

Searching
The engine is searching for product images.

Details
For each status, shows the number of steps in that status. Also displays the total number of steps.

For information about each individual step, double-click the step to open the Step window.

The Step window
If you double-click a step in the Step List window, The Step window opens. It has three tabs:

Status tab:
The Status tab shows the status of the installation step (Ready, Success, Error, or Held). You can change the status from Error to Ready if the condition that caused a step to fail has been removed. This is an example of the tab:
Properties tab:
The **Properties** tab gives the user parameters required by the step. These might be the parameters you have input in the installation wizard, or values that the wizard has determined according to the logic of its operations. This is an example of the tab:

![Figure 14. Step status tab](image)

For example, in this tab the property **DB2 Client Flag** is an internal property determined by the wizard.

The properties are of three types:

A **command**

Some of the steps have properties that include a command string. *This must never be edited.* The command string has positional parameters and generated parameters. If you change even one character the command might fail.

**Editable parameters**

These are parameters that are used by the step, but can be edited by you. The name of the parameter is the same as the name used on the wizard panel when you input the data.
Internal parameters
These are parameters generated by the wizard. They are recognizable because you did not input them in the wizard panels. For example, the DB2 Client Flag in the above example screen is an internal flag which tells the step whether it is to install the DB2 Server or DB2 Client. These must never be edited. They might be linked to other parameters in a way that is not obvious to you.

Output tab:
The Output tab shows the output and any errors that occurred for the installation step, and also the commands that were performed by the installation. This is an example of the tab:
The Output tab has the following entries:

**Time Stamp**
- The time that the command was run.

**Return code**
- The return code for the operation:
  - 0 OK
  - 1 - 9 Error
DiagRecord
A unique point of failure identification. This can be quoted to IBM Software Support if you need to request assistance.

Command
The command that failed.

Command output
Any output from the command (such as a return code or an error message number)

Error log
Shows a list of errors that occurred during the installation of the step.

**Correcting a failed step and continuing the installation**
To correct a failed step and continue the installation, use the following procedure:

1. Use the **Output** tab to determine what problem occurred.
2. Consult the sections in this guide that describe how to resolve problems found with the installation or the help for the error message that has been displayed.
3. If the solution to the problem requires you to change one of the values that you entered in the installation wizard, consult "Deciding whether to resume the wizard or rerun it" on page 187 and determine from the guidelines there, if you should rerun the wizard from scratch, or if it is appropriate to correct the value and resume the wizard. If the latter, follow this procedure:
   a. Select the **Properties** tab and make the required changes. Click **Apply**.
      The error description might make reference to a property that is not available for editing in the step that failed. In this case you must do the following:
      1) Close the Step window.
      2) Double-click the preceding step and check if the **Properties** tab contains the property you require. If it does not, then close the Step window and try the next preceding step; continuing until you locate the property to change.
         When you find the property, change its value and then continue with the rest of the steps in this procedure, from the step that you have modified, not from the step that failed.
   b. Double-click each of the other steps in the installation in turn and click the **Properties** tab for the step. If the step includes the property whose value you changed above, change the value of the property accordingly and click **Apply**. This is necessary because the step properties are not interlinked.
   c. For each modified step, lick the **Status** tab, change the Status to **Ready**, then click **Apply**. The Step list is redisplayed.
4. If instead, the solution to the problem does not require you to change any of the values that you entered in the installation wizard, resolve the problem outside the wizard, then change the Status to **Ready** and click **Apply**. The Step list is redisplayed.
5. Determine which is the earliest step in **Ready** status.
6. If you want to run just the first step in **Ready** status, to ensure, for example, that the change you made has worked, click **Run next**. This runs the first step in the step list (in step number order) with a status of **Ready**. When the step finishes successfully you run the other steps in the installation in the same way, in sequence, or use **Run all**.
7. To resume the installation in one go, click **Run all**. The wizard attempts to complete all outstanding steps, starting with the first step in **Ready** status.
Deciding whether to resume the wizard or rerun it

The fact that the wizard has a facility that allows you to diagnose the problem, correct it, and resume it, does not mean that you must do so. There are a number of scenarios when it is quicker to rerun the wizard, or you are more sure of success by doing so. This section helps you to decide which is the best action to take.

**Note:** Diagnosing and resuming a failed installation is a process that must be guided, either by following the instructions in the sections that follow in this manual, or by following instructions from IBM Software Support.

The facility to diagnose, correct, and resume a failed installation can be very useful for you, but if it is not done correctly can require more work than rerunning it.

The following sections detail different installation scenarios and suggest the best way to proceed.

**Installing an agent or the command line client:**
If you are installing an agent or the command line client, it is always easier to rerun, rather than resume, a failed installation. This is because the steps are few, and can all be rerun.

**Installing a master domain manager, a backup master domain manager, or the connector:**
If you are installing a master domain manager, a backup master domain manager, or the connector, you must follow these guidelines:

**Reason for failure:**
Consider the reason for failure and what is needed to fix the problem:

**External reason**
If the wizard has failed for an external reason, that you can correct, you can always resume from the failed step.

For example, in an installation of the master domain manager, the database support that you chose in the wizard must be running during the installation. When you supply the information to identify the RDBMS instance, the wizard checks that it is running and gives an error if not. However, if the RDBMS support stops running for any reason, after the wizard has checked it is running, but before the wizard starts to install the Tivoli Workload Scheduler database, the wizard stops. To resolve the problem, restart the RDBMS support and resume the installation from the failed step.

**Non-valid installation data**
If the wizard fails because data supplied to the wizard is not valid, you must consider which data is not valid:

<TWS_user> ID or password
If there are any problems with the <TWS_user> ID or password, you must quit the installation and rerun it. Many of the steps have the <TWS_user> as a property, and because many crucial factors in the installation are linked to the <TWS_user>, you must not try to change the ID and then resume.

**Installation directory**
Like the <TWS_user> ID, this is important to the installation. If the supplied value has some problem you must quit the installation and rerun it.
Ports
The ports used by the embedded WebSphere Application Server are checked at the moment you input them, but if one of them becomes busy by the time the wizard starts to configure the embedded WebSphere Application Server, the installation stops. In this case, you can proceed to change the value of the port being used in the step and resume the installation, because the ports are only used in one step.

Database data
The data relating to the configuration of the RDBMS support and the installation of the Tivoli Workload Scheduler database might be used in any or all of the database-related steps. Look at the names of the steps to determine which they are. If you change a value in one, open them all and check if the value is used in others.

Other data
For all other installation data, check every step to determine where the data item is being used.

If in doubt, rerun the installation.

Where is the problem:
Follow these guidelines depending on where, in the installation, the problem occurs:

Early steps
Generally, if the problem occurs in one of the early steps, it is almost as quick, and generally more reliable, to rerun the installation, than correct the data and resume it.

After the database has successfully been installed
If the problem occurs after the database has been successfully installed, but you want to rerun the installation, or resume from some point before the database configuration steps, there is no need to drop or uncatalog the database because the wizard finds the existing database and continues.

Cleaning up before a rerun:
If you decide to rerun, there should be no need to clean up anything. All the data structures that are installed can be overwritten.

Deciding whether to resume immediately or exit and resume later
If you decide to diagnose, correct, and resume the wizard, you can choose to do it immediately, or to quit the installation and resolve the problem later:

Diagnose failure
If you choose to diagnose the failure immediately, the Step List window is opened. See “The Step List window” on page 181 for more details.

Quit installation
If you select to quit the installation, a summary of the progress of the installation is displayed, and the InstallShield wizard is closed.

You can discover the reason for the failure by looking in the installation log files, correct the problem, and later perform a restart of the installation using the resume option, see “Stopping and resuming an interactive installation” on page 189.

Note these considerations:
• Do not close the panel by clicking on the Close icon: ✗. If you do, the wizard is unable to save the troubleshooting information that you need for the resume. Instead, if you are sure you want to quit the installation, click Quit installation.

• You can only resume the last installation of each SETUP attempt regardless of the component it was installing. Every time you click Next on the Summary panel when running an installation of any component, any previous troubleshooting information about a previous installation of any component is overwritten. If you want to be able to resume multiple installations on the same computer, back up the entire <tempDir>/TWA/tws86 directory after each installation attempt has stopped, and then, for each installation you need to resume, restore this data from the backup and resume the installation.

Stopping and resuming an interactive installation
You can stop and resume an installation at any time. For example:
• The installation is running successfully but you want to pause it and resume it later
• The installation has failed and you want to reboot the computer to correct a problem

To stop an interactive installation that is running click Stop. The wizard asks you if you want to quit the installation after the current step is completed. If you reply Yes, the installation completes the step being performed and then displays a summary panel of the completed activities. Click Finish to exit.

To stop an installation that has just failed, select Quit installation, click Next, and confirm your decision.

To stop an installation that is on the Step List window, click Cancel, and confirm your decision.

To resume the installation, enter the following command:

<setup_file_name> -resume

where <setup_file_name> is one of the following:

Windows operating system
   setup.exe

UNIX and Linux operating systems
   SETUP.bin

The InstallShield wizard recognizes that a previous installation has not completed, and the Step List window opens. From here you can continue the previous installation at the point where it stopped. If the steps in the Step List window have no errors, you can resume the installation; otherwise you must correct the error that caused the installation steps to fail, before resuming those steps. See “The Step List window” on page 181 for details.

Example procedure for resolving a problem
This section describes an example procedure for resolving a problem and finishing the installation.

Assume that you are upgrading an existing instance of Tivoli Workload Scheduler version 8.3, using DB2, and the “Configure the Tivoli Workload Scheduler instance”
step has failed. This could be for a number of reasons. The procedure for resolving the problem starts when the installation stops and the Diagnose Failure window opens (see Figure 12 on page 180).

1. On the Diagnose Failure window, select Diagnose failure and click Next. The Step List window opens (see Figure 13 on page 181).
2. Double-click the step that failed, in this case the "Configure the Tivoli Workload Scheduler instance" step.
3. Click the Output tab (see Figure 16 on page 185) and determine the cause of the problem.
4. Fix the problem. For this scenario it is assumed that the workstation name that you originally supplied to the installation wizard is not valid. Click the Properties tab and change the workstation name to a valid value. Click Apply.
5. Change the Status on the Status tab (see Figure 14 on page 183) to Ready, and click Apply. The Step List window opens again. This time the status of all the steps yet to be performed is set to Ready.
6. Double-click the other steps in turn and click their Properties tab. If you find the workstation name field, change the value as you did for the failed "Configure the Tivoli Workload Scheduler instance" step. Click Apply.
7. In this case, the steps that you changed have not yet been run, so the earliest step you changed is also the step that failed. The status of this step is already set to "Ready", so there is nothing further to do.
8. When you have checked the properties for the affected steps, click Run all. The installation wizard resumes and completes the installation.

**Recovering a failed silent InstallShield wizard installation**

If the silent wizard stops, follow this procedure:

1. Open the installation log and establish at what point the installation failed. The location of the installation log files is described in "Installation log files" on page 32.

The installation is performed in two distinct phases:

**Validation phase**

The input parameters are validated, but no installation action is taken. In the log file the validation of the input parameters is indicated by the action: validateFields.

**Step execution phase**

The installation is performed in a series of steps. In the log, each step starts with a message that begins "Running step:"

2. When you have discovered what the problem is, and in what phase it started, determine how to resolve it. You might have to correct a parameter or change something in the installation environment (for example, create more disk space).

3. When you have resolved the problem, you can rerun or resume the wizard. To determine which you should do, see "Deciding whether to resume the wizard or rerun it" on page 187. Follow these instructions for how to proceed after making the decision:

**Rerun the wizard**

You must rerun the wizard if the error was found in the Validation phase.

If the wizard was in the Step execution phase you can always rerun it, but you must consider that the wizard attempts to redo each step.
Thus, you might need to clean up the installation environment after the failed installation before rerunning it.

If you need to change an input parameter, edit the response file. Then rerun the wizard, just reissuing the silent wizard command as you issued it originally.

**Resume the wizard**

*You can only* resume the wizard if the error was found in the *Step execution phase*. The resume option uses the interactive wizard. You cannot resume the wizard silently, because an interaction is required to resume the failed step.

To resume the wizard, reissue the silent wizard command as you issued it originally, with these changes:

- Add the parameter `-resume`
- Remove the parameter `-silent` when you ran it originally. If you do not remove this parameter, the installation cannot resume.

The Step list window of the interactive wizard is displayed, where you can optionally change the values of the data input parameters, and resume the installation at the failed step. See ["The Step List window"](page 181,) and follow the instructions in that section.

**Recovering a failed upgrade**

In the case of a failed upgrade, collect any logs or error messages and contact IBM Software Support.

**Analyzing return codes for Tivoli Workload Scheduler for Additional Plug-ins silent installation**

Check the error and warning messages issued by Tivoli Workload Scheduler for Additional Plug-ins, during the silent installation process.

This section lists the *errors* and the *warnings* messages returned by InstallAnywhere during the silent installation process.

The *errors* and *warnings* are organized into two tables:

- Default InstallAnywhere error messages, [Table 15 on page 192](#)
- Additional plug-in installation error messages, [Table 16 on page 193](#)

When running the installation in silent mode, no messages are displayed. The messages are written in the silent installation log files listed in [Table 9 on page 111](#).

If the response file you specify in the command line does not exist or the file name is incorrect, the silent installation process does not write the log files. To have the correct return code for the silent installation process, issue:

```
start /w <silent command>
```

where:<code>&lt;silent command&gt;</code> is the command you launch to perform silent installation.

You can check the error codes found in the installation log files during the silent installation process, with the codes in the following tables to obtain the specific description of the error message: [Table 15 on page 192](#) shows the default InstallAnywhere error messages written in the log files during the silent installation execution.
<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Success: The installation completed successfully without any warnings or errors.</td>
</tr>
<tr>
<td>1</td>
<td>The installation completed successfully, but one or more of the actions from the installation sequence caused a warning or a non-fatal error.</td>
</tr>
<tr>
<td>8</td>
<td>The silent installation failed because there is an error in one or more installation steps.</td>
</tr>
<tr>
<td>-1</td>
<td>One or more of the actions from the installation sequence caused a fatal error.</td>
</tr>
<tr>
<td>1000</td>
<td>The installation was canceled by the user.</td>
</tr>
<tr>
<td>1001</td>
<td>The installation includes an invalid command-line option.</td>
</tr>
<tr>
<td>2000</td>
<td>Unhandled error.</td>
</tr>
<tr>
<td>2001</td>
<td>The installation failed the authorization check. It might indicate an expired version.</td>
</tr>
<tr>
<td>2002</td>
<td>The installation failed a rules check. A rule placed on the installer itself failed.</td>
</tr>
<tr>
<td>2003</td>
<td>An unresolved dependency in silent mode caused the installer to exit.</td>
</tr>
<tr>
<td>2004</td>
<td>The installation failed because not enough disk space was detected while running the Install action.</td>
</tr>
<tr>
<td>2005</td>
<td>The installation failed while trying to install on a Windows 64-bit system, because the installation does not include support for Windows 64-bit systems.</td>
</tr>
<tr>
<td>2006</td>
<td>The installation failed because it was launched in a UI mode that is not supported by this installer.</td>
</tr>
<tr>
<td>3000</td>
<td>Unhandled error specific to a launcher.</td>
</tr>
<tr>
<td>3001</td>
<td>The installation failed due to an error specific to the lax.main.class property.</td>
</tr>
<tr>
<td>3002</td>
<td>The installation failed due to an error specific to the lax.main.method property.</td>
</tr>
<tr>
<td>3003</td>
<td>The installation was unable to access the method specified in the lax.main.method property.</td>
</tr>
<tr>
<td>3004</td>
<td>The installation failed due to an exception error caused by the lax.main.method property.</td>
</tr>
<tr>
<td>3005</td>
<td>The installation failed because no value was assigned to the lax.application.name property.</td>
</tr>
<tr>
<td>3006</td>
<td>The installation was unable to access the value assigned to the lax.nl.java.launcher.main.class property.</td>
</tr>
<tr>
<td>3007</td>
<td>The installation failed due to an error specific to the lax.nl.java.launcher.main.class property.</td>
</tr>
<tr>
<td>3008</td>
<td>The installation failed due to an error specific to the lax.nl.java.launcher.main.method property.</td>
</tr>
<tr>
<td>3009</td>
<td>The installation was unable to access the method specified in the lax.nl.launcher.java.main.method property.</td>
</tr>
<tr>
<td>4000</td>
<td>A Java executable could not be found at the directory specified by the java.home system property.</td>
</tr>
<tr>
<td>4001</td>
<td>An incorrect path to the installer jar caused the relauncher to launch incorrectly.</td>
</tr>
</tbody>
</table>
Table 16 shows the error messages issued during Tivoli Workload Scheduler for Additional Plug-ins silent installation of the plug-ins.

Table 16. InstallAnywhere error messages for additional plug-ins

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>The required parameter does not contain a value.</td>
</tr>
<tr>
<td>12</td>
<td>The file specified in response file does not exist.</td>
</tr>
<tr>
<td>13</td>
<td>The plug-in file specified is not a zip file.</td>
</tr>
<tr>
<td>14</td>
<td>The plug-in you specified does not contain the plugin.xml file.</td>
</tr>
<tr>
<td>15</td>
<td>The installation process does not find a Tivoli Workload Automation instance on this system.</td>
</tr>
<tr>
<td>16</td>
<td>You cannot perform the action you specified on the selected instance.</td>
</tr>
<tr>
<td>17</td>
<td>You are performing the installation on a workstation that does not have enough disk space.</td>
</tr>
<tr>
<td>18</td>
<td>The path you specified does not contain a valid installation of Tivoli Workload Scheduler.</td>
</tr>
<tr>
<td>19</td>
<td>The operating system, where you are performing installation, is not supported.</td>
</tr>
<tr>
<td>20</td>
<td>The plug-in you specified contains a plugin.xml file with syntax errors.</td>
</tr>
<tr>
<td>21</td>
<td>The plugin.xml file you specified, lists some files that are not contained in the plug-in.</td>
</tr>
<tr>
<td>22</td>
<td>You do not specify a plug-in in the response file.</td>
</tr>
<tr>
<td>23</td>
<td>An higher version of the selected plug-in is already installed on this instance.</td>
</tr>
<tr>
<td>24</td>
<td>The plug-in (zip file) you are installing does not contain the required jar file or the jar file is not located in the correct path.</td>
</tr>
<tr>
<td>25</td>
<td>You cannot install the selected plug-in using the current plug-in installer version.</td>
</tr>
<tr>
<td>26</td>
<td>The plug-in (zip file) you are installing does not contain the required licences or the licences are not located in the correct path.</td>
</tr>
<tr>
<td>27</td>
<td>The ACTION_TYPE parameter value specified in the response file, must be DEPLOY or UNDEPLOY.</td>
</tr>
<tr>
<td>28</td>
<td>You cannot accepted the license agreement in the response file.</td>
</tr>
<tr>
<td>29</td>
<td>The installation process cannot save the updates for TWARegistry.dat file.</td>
</tr>
<tr>
<td>30</td>
<td>The installation process is unable to update the config.ini file.</td>
</tr>
<tr>
<td>31</td>
<td>The installation process is unable to copy plug-in files on the target system.</td>
</tr>
<tr>
<td>32</td>
<td>Cannot install the selected plug-in on the Tivoli Workload Scheduler instance because the Java extension is not installed</td>
</tr>
</tbody>
</table>

Problem scenarios: install, reinstall, upgrade, migrate, and uninstall

This section contains known problem scenarios that could occur with the install, reinstall, upgrade, migrate, and uninstall of Tivoli Workload Scheduler components. It is divided into these topics:

- “Problems installing on Windows operating system” on page 194
- “Problems installing on UNIX” on page 200
Problems installing on Windows operating system

The following sections describe problems that could occur when installing on Windows, and their workarounds:

- “An installation on Windows fails.”
- “Installation fails because the host name is truncated” on page 195
- “An InstallShield wizard installation fails with the message “CMW3202E Command failed.”” on page 195.
- “The InstallShield wizard commit step fails on Windows with AWSDEQ024E error” on page 195.
- “The installation on Windows receives the warning AWSGAB005W” on page 195.
- “On a Windows 2003 domain, the application server installation fails with an apparent credentials problem but the credentials are correct” on page 195.
- “The user account is not created on Windows 2000 with system error 56b” on page 196.
- “The user account is not created on Windows - create it manually” on page 196.
- “The Windows services fail to start after installation” on page 198.
- “The Tivoli Workload Scheduler services fail to start after installation” on page 198.
- “Installation fails with error “SQL1219N - The request failed because private virtual memory could not be allocated.”” on page 198.

An installation on Windows fails:
You have tried to install the product using the InstallShielrd wizard on a Windows workstation, but the installation has failed.

Cause and solution

A possible cause of a failure of an InstallShield wizard installation on Windows is that the Services window of the Administrative Tools in the Control Panel is open. This is because, with the window open, the Services registry cannot be updated.

Close the Services window and do one of the following:

- If you have selected the wizard option to recover a failed installation by rerunning the step that failed, you can now set the status of that step to “Ready” and resume the step (see “Recovering a failed interactive InstallShield wizard installation” on page 179).
If you have completely stopped the installation, you can resume it (see “Stopping and resuming an interactive installation” on page 189).

Installation fails because the host name is truncated:
You are installing a Tivoli Workload Scheduler component with the InstallShield wizard on a system where you have defined a host name longer than 15 characters. The installation fails because of a host name mismatch.

Cause and solution

This problem is caused by Windows. When you define a host name Windows places no limit on the length of the host name. However, if you supply a host name longer than 15 characters, when you reboot the Windows system to make the host name active, Windows truncates the host name to 15 characters, and logs a message in the system log. If you do not notice that message, you will not know that the name has been truncated, and will supply the full version of the host name on the installation panel.

Unfortunately, the Windows Java classes that use that supplied long host name do not truncate the name, so report a mismatch.

To resolve the problem, do one of the following:

- Specifically rename the host name to 15 characters or less, reboot the computer and restart the installation
- If the truncated name is acceptable, use the step restart facility (see “Recovering a failed interactive InstallShield wizard installation” on page 179 or “Recovering a failed silent InstallShield wizard installation” on page 190) to modify the supplied host name to its truncated form, and continue the installation.

An InstallShield wizard installation fails with the message "CMW3202E Command failed."
You are running an InstallShield wizard installation on Windows and the wizard fails in the step "Configure the Tivoli Workload Scheduler instance", with error messages like the following:

"atinstall.exe"
CMW3202E Command failed.

Cause and solution

A possible cause for this problem is that a file (not a directory) exists on the installation drive with the name <drive_letter>\program (for example, D:\program. Delete or rename this file and resume the installation at the failed step.

The InstallShield wizard commit step fails on Windows with AWSDEQ024E error:
You are installing Tivoli Workload Scheduler on Windows with the InstallShield wizard, interactive or silent. The commit step fails, and the error log includes a message similar to the following:

AWSDEQ024E Error owner is not of type user in :
"makesec failed with error code: 1"

Cause and solution
The problem is probably caused by a non-valid setting of the TMP environment variable. For example, the above version of message AWSDEQ024E is produced if the value of the TMP variable is set to C:\tmp instead of C:\tmp.

To resolve the problem, quit the installation wizard, uninstall Tivoli Workload Scheduler and rerun the installation wizard. You cannot just correct the TMP variable and resume the installation. If you have problems running the uninstallation, see “Uninstalling Tivoli Workload Scheduler manually” on page 236.

The installation on Windows receives the warning A W S G A B 0 0 5 W:

An installation on Windows receives the following error:

| AWSGAB005E The account cannot be verified automatically. > |
| Check that the supplied login and password are valid and > |
| satisfy the local security policy. |

**Cause and solution**

This is a known Windows limitation, dependent upon the security settings of your Windows workstations. The password you supplied for the <TWS_user> might be perfectly valid, but the installation wizard is unable to validate it.

The problem does not block the installation. If you are using the interactive wizard, the installation displays this message; giving you the option to click Next to continue. If you are using the silent option, the installation goes ahead.

To resolve the problem, take the following steps:

1. After the installation has completed, check whether the supplied password was correct. If it was, you need do no more.
2. If the password was incorrect, when the product tries to start the Windows services it has created, the services fail. Use the Windows facilities to modify the password used by these services so that it matches the password of the <TWS_user>. To do this perform these steps for each service:
   a. In the Windows Services panel, locate the Tivoli Workload Scheduler service with the incorrect password.
   b. Access the properties of this service, for example by right-clicking the service and selecting Properties.
   c. Click the Log On tab.
   d. Enter and confirm the correct values for the password.
   e. Start the service.

**On a Windows 2003 domain, the application server installation fails with an apparent credentials problem but the credentials are correct:**

You are installing on a Windows 2003 domain any Tivoli Workload Scheduler component that also installs the application server, but the installation fails at the step where the application server is being installed. The error message given is:

```
com.ibm.websphere.security.auth.NSLoginFailedException: Authentication failed for user mdm84 with the following error message Logon failure: unknown user name or bad password.
```

When you check the credentials you find that the User ID and password are correct.

**Cause and solution**
A possible cause for this problem is that you have not installed SP2 on Windows 2003. Without SP2, Windows has a known bug that when it sets the "impersonate a client after authentication" right, it also deletes the network connection icon, so that the installation cannot communicate with the application server. SP2 is a prerequisite of the installation.

Do the following:
1. Install Windows 2003 SP2
2. Make sure that the 'impersonate a client after authentication' right is applied not only to the <TWS_user> but also to the Administrators group, and the Service system account.
3. Rerun the installation.

The user account is not created on Windows 2000 with system error 56b:
On Windows 2000 operating systems, you are running an InstallShield wizard installation specifying a domain user (domain\user) as the <TWS_user>. The installation fails at the user account creation stage giving the following message:
WARNING: USER DOES NOT EXIST #System error <56b>

Cause and solution

The problem is actually caused by a synchronization error between the Windows 2000 domain server and client. The account creation process and the process that adds the account to the appropriate user group are not synchronized, giving the impression that the account creation has failed.

Normally, by the time you have noted the error, the account creation has been completed. So, to resolve this problem, just set the user account creation step of the installation to "ready" and continue the installation.

However, if the process fails for a second time, you must create the user manually, and then resume the failed step.

The user account is not created on Windows - create it manually:
On Windows operating systems, the installation automatically creates the Tivoli Workload Scheduler user with the appropriate rights, if the user does not already exist. However, if the installation encountered problems with the creation of the user, you can perform the following steps.
1. Back out of the installation.
2. Create a local user account with a name of your choice on the workstation where you want to install Tivoli Workload Scheduler.

Note: You can also use an existing user account. Ensure, however, that this user is a member of the Windows Administrators group.
3. Grant this <TWS_user> the following advanced user rights:
   - Act as part of the operating system
   - Increase quotas
   - Log on as batch job
   - Log on as a service
   - Log on locally
   - Replace a process level token
4. Rerun the installation, citing the name of the account you created when requested.
The Windows services fail to start after installation:
If the Tivoli Token service and the Tivoli Workload Scheduler for <TWS_user> services fail to start after installation, there was probably a problem with the <TWS_user> password. This scenario is described in “The Tivoli Workload Scheduler services fail to start after installation.”

The Tivoli Workload Scheduler services fail to start after installation:
On Windows, both the Tivoli Token service and the Tivoli Workload Scheduler for <TWS_user> service (batchup) fail to start for the first time (after a successful installation).

Cause and solution

During the installation you selected the option to create the <TWS_user>, but an error was found with the password you supplied (for example, the supplied password might not have satisfied the security policy on the workstation). You used the restart facility (see “Recovering a failed interactive InstallShield wizard installation” on page 179) to recover the situation. On the recovery panel you entered a different value for the password than the value entered originally. This value was valid, so the program went ahead and completed the installation.

However, during the completion of the installation, the <TWS_user> was created using the new password that you entered on the recovery panel, while the services were created using the original password value.

The reason for this is that when the installation wizard starts the installation steps, after having accepted the input of all of the installation variables, it creates each of the steps complete with the properties (variables) required to perform the step. If a step fails, and the Step List window opens (see “The Step List window” on page 181), the properties that are displayed for one step are quite separate from the properties of another step. Thus, if you changed the password for one step (for example, the “Create the <TWS_user> (Windows only)” step), you must have also changed it for all the other steps where it is used (for example, the “Configure the Tivoli Workload Scheduler instance” step (see “Correcting a failed step and continuing the installation” on page 186).

To resolve this problem after the installation has completed, you must change the password for the services to the value that you entered on the recovery panel, following the procedure described in Tivoli Workload Scheduler: Administration Guide.

If you become aware of this problem before the installation is complete, you can choose either to let the installation go ahead and change the password afterwards, as described in the previous paragraph, or to exit from the installation, uninstall whatever you have installed, following the procedures described in the Tivoli Workload Scheduler: Planning and Installation Guide, and rerun the installation.

Installation fails with error “SQL1219N - The request failed because private virtual memory could not be allocated.”:
During the master domain manager installation of DB2 on Windows, the installation fails with the error “SQL1219N - The request failed because private virtual memory could not be allocated.”

Cause and solution
The request failed because private virtual memory could not be allocated. This can occur because the DB2 administrator is not part of the Administrators group. You must add the DB2 administrator to the Administrators group by performing the following steps:
1. Cancel the installation wizard.
2. Add the DB2 administrator, for example db2admin, to the Administrators group.
3. Restart the workstation.
4. Resume the installation.

**Problems installing on AIX**
The following problems could occur while installing on AIX:
- "ISMP installation on AIX hangs"
- "ISMP installation on AIX 6100-05 hangs"

**ISMP installation on AIX hangs:**
The installation hangs while using the ISMP wizard in interactive mode.

**Cause and solution**
If this happens, do the following:
1. Cancel the interactive installation.
2. Follow the instructions provided in "Recovering a failed interactive InstallShield wizard installation" on page 179 to back out any installation actions that have already been performed.
3. Perform the installation in silent mode.
   Instructions for performing a silent ISMP installation are provided in Chapter 3 of *Tivoli Workload Scheduler: Planning and Installation Guide*.

**ISMP installation on AIX 6100-05 hangs:**
You are installing on AIX 6100-05, and receive a message indicating that there is a problem validating the authentication information (userid and password).

**Cause and solution**
You receive the following error:

```
<2011.05.04 14:14:32 - Configure the Tivoli Workload Scheduler database (nc118226)>
Installation completed with errors <<<<-
/tmp/TWA/tws86/scripts/createdb_root.sh
TWS86 TRUE TWS86 ND nc118227.romelab.it.ibm.com 50000 db2inst1
******* DUMMY FALSE FALSE FALSE
<OUT [AWSJIS038E An internal error has occurred. An unspecified internal error has occurred during the installation process. Checking node
Checking for database existence
Error in attaching to the node TWS86 ND.
Check out the node TWS86 ND in the DB2 node catalog and the supplied authentication data (username/password)
CMW3202E Command failed. RC: 3]
ERR [AWSJIS031E An internal error has occurred. An internal program has failed.]
```

The authentication information (userid and password) are correct. To solve the problem perform the following steps:
1. In the /tmp/TWA/tws86/scripts/createdb_root.sh script, modify the following line:
su - $DB2_ADMINISTRATOR -c "cd $TWS_TEMPDIR/scripts && ./dbsetup.sh $TWSDBNAME $TWSCLIENTFLAG $1 $2 $3 $4 '$5' $6 $7 $8 $9"

as follows:

su - $DB2_ADMINISTRATOR -c "source .profile && cd $TWS_TEMPDIR/scripts && ./dbsetup.sh $TWSDBNAME $TWSCLIENTFLAG $1 $2 $3 $4 '$5' $6 $7 $8 $9"

2. In the /tmp/TWA/tws86/DWB/scripts/createdb_root.sh script, modify the following line:

su - $LOCAL_DB2_USER -c "cd $SCRIPT_DIR && ./dbmigrate.sh $1 $2 '$3' $4 $5"

as follows:

su - $LOCAL_DB2_USER -c "source .profile && cd $SCRIPT_DIR && ./dbsetup.sh $1 $2 '$3' $4 $5"

3. Run the failed step again.

**Problems installing on UNIX**

The following problems could occur:

- "The installation fails on UNIX with a problem validating the Java Virtual Machine"
- "An incorrect password is supplied for the TWS_user on UNIX"

**The installation fails on UNIX with a problem validating the Java Virtual Machine:**

You are installing on UNIX, and receive a message indicating that there is a problem validating the Java Virtual Machine (JVM).

**Cause and solution**

This might be caused by a timeout problem. The InstallShield wizard uses a default timeout of five seconds during its operations to validate the version of JVM that you have installed. For a variety of reasons this might be insufficient.

Relaunch the installation wizard (interactive or silent) adding the following parameter:

```
is:jvmtimer 10
```

This extends the timeout to 10 seconds, and if this does not work, you can try extending it to 20 seconds. If the JVM still does not validate correctly, contact IBM Software Support for assistance.

**An incorrect password is supplied for the TWS_user on UNIX:**

You have supplied an incorrect password for the TWS_user on UNIX. An appropriate error message is displayed, but it is displayed during the stepped installation rather than when you input the password. You need to determine how to recover.

**Cause and solution**

The TWS_user password cannot be checked on UNIX at time of input, for technical reasons. If an incorrect password is provided, the error is not discovered until the wizard tries to install the product, after it has already successfully installed the Tivoli Workload Scheduler database, and the embedded WebSphere Application Server.
To recover from this situation, do the following:

1. Quit the wizard
2. Delete the application server directory: ${WAS_HOME}/eWAS/, where ${WAS_HOME} is the environment variable that contains the installation path of the embedded WebSphere Application Server
3. Rerun the wizard
4. Supply the correct password for the TWS_user, when requested

Problems installing on HP-UX
The following problems could occur:

- "An InstallShield wizard installation cannot start on HP-UX"
- "An InstallShield wizard installation fails on HP-UX with an error installing the bundled JRE"
- "An InstallShield wizard installation fails on HP-UX with a "run error""

An InstallShield wizard installation cannot start on HP-UX:
You are trying to install Tivoli Workload Scheduler on HP-UX using the InstallShield wizard. The wizard does not start.

Cause and solution
This is probably due to insufficient threads being available to the installation program.

Set the max_thread_proc kernel parameter to a minimum of 128 so that the installation can start.

See [http://www.ibm.com/support/docview.wss?rs=672&uid=swg27012175](http://www.ibm.com/support/docview.wss?rs=672&uid=swg27012175) for details of the typical kernel parameters to use to run Tivoli Workload Scheduler on HP-UX.

An InstallShield wizard installation fails on HP-UX with an error installing the bundled JRE:
You are installing on HP-UX where the required level of JRE is not installed. The installation wizard tries to install the bundled JRE but fails. The following message is received:

Bundled JRE is not binary compatible with host OS/Arch or it is corrupt.
Testing bundled JRE failed.

Cause and solution
This problem is probably caused by the HP-UX configuration parameter MAXDS1Z having been set to a value that is too low. Set the MAXDS1Z configuration parameter to a minimum of 128 MB, and retry the installation.

An InstallShield wizard installation fails on HP-UX with a "run error":
You are trying to install Tivoli Workload Scheduler using the InstallShield wizard on an AIX or HP-UX operating system. The installation fails giving the following exception in thread "main":

```
java.lang.NoClassDefFoundError: run error
```

This problem is described in "An InstallShield wizard installation fails on AIX or HP-UX with a "run error"" on page 210.
Problems installing on Oracle Solaris

The following problem could occur:

An installation fails on Oracle Solaris with the error "The command line parameter, -installRoot, is invalid":
The installation of a component on Oracle Solaris fails. The following error messages are given:
AWSJIS038E: An unspecified internal error has occurred during the installation process.
ERROR: The command line parameter, -installRoot, is invalid
Use -usage to see the available command line options
ERROR installing WAS Express, check system stderr/stdout

Cause and solution

The problem is possibly caused by an incorrect PATH environment variable, which has the search path relating to an X/Open specification, for example XPG4, in the incorrect order.

Consult the Oracle Solaris documentation and support website and ensure that the PATH variable is correctly expressed. Correct any error you find and retry the installation.

An installation fails on Oracle Solaris with the error "SQL0101N The statement is too long or too complex. SQLSTATE=54001":
The installation of a component on Oracle Solaris fails. The following error message is given in the DB2 /export/home/db2admin/sqllib/db2dump/db2diag.log:
SQL0101N The statement is too long or too complex. SQLSTATE=54001

Cause and solution

The problem is possibly caused by an incorrect kernel parameter on Solaris.

To solve the problem perform the following steps:
1. Save in another directory for example home_dir the content of the /tmp/TWA/tws86 directory.
2. Run db2osconf to see the kernel parameter settings and set the value suggested by the command. Consult the IBM DB2 documentation which describes how to modify kernel parameters on Oracle Solaris.
3. Restore the content of the /tmp/TWA/tws86 directory in the /tmp/TWA/ directory.
4. Perform the setup -resume command to recover the installation.

An installation fails on Oracle Solaris with the error "SQL1084C Shared memory segments cannot be allocated. SQLSTATE=5701":
The installation of a component on Oracle Solaris fails. The following error message is given in the DB2 /export/home/db2admin/sqllib/db2dump/db2diag.log:
SQL1084C Shared memory segments cannot be allocated. SQLSTATE=5701

Cause and solution

The problem is possibly caused by an incorrect kernel parameter on Solaris.

To solve the problem perform the following steps:
1. Save in another directory for example home_dir the content of the /tmp/TWA/tws86 directory.
2. Run `db2osconf` to see the kernel parameter settings and set the value suggested by the command. Consult the IBM DB2 documentation which describes how to modify kernel parameters on Oracle Solaris.

3. Restore the content of the `/tmp/TWA/tws86` directory in the `/tmp/TWA/` directory.

4. Perform the `setup -resume` command to recover the installation.

**Problems installing on Linux**

The following problems could occur:

- **An InstallShield wizard installation fails on Linux with an error installing the bundled JRE**:
  
  You are installing on Linux where the required level of JRE is not installed. The installation wizard tries to install the bundled JRE but fails. The following message is received:

  ```
  This application requires a Java Run Time Environment (JRE)
to run. Searching for one on your computer was not successful.
Please use the command line switch -is:javaxome to specify
a valid JRE. For more help use the option -is:help.
  ```

  **Note:** The solution indicated in this InstallShield wizard message probably does not work.

  **Cause and solution**

  The probable cause is that the `bc` utility is a prerequisite of the InstallShield wizard, but is not installed by default on all Linux platforms (on Red Hat Linux, version 2.1, for example, it is included only in Service Pack 2).

  To check for the existence of the utility run this query on the rpm registry: `rpm -q bc`

  If the utility is missing, consult your operating system’s support resources to determine how to obtain it. When it is successfully installed, rerun the installation.

- **A non-English installation on Linux finishes correctly, but the start of Tivoli Workload Scheduler gives one or more errors**:

  You installed a non-English version of Tivoli Workload Scheduler on Linux, but when the product starts errors are given.

  **Cause and solution**

  The problem might be the code page of the workstation. To support languages other than English, Tivoli Workload Scheduler requires the code page to be `UTF8`. Reset the code page and restart the product and you should have no reoccurrence of this problem.
Java Virtual Machine (JVM) failure when installing on a Red Hat Enterprise Linux (RHEL) Version 5 or a Suse Linux system Version 11:
Problem description:
When working with Tivoli Workload Scheduler on a Red Hat Enterprise Linux Version 5 or a Suse Linux system Version 11, you might receive the error "Failed to find VM - aborting".

Cause and solution
Linux systems have a new security feature named 'Security Enhanced Linux', or SELinux for short. A weaker version of SELinux was included in Red Hat Enterprise Linux Version 4, and was disabled by default. On these versions of Red Hat Enterprise Linux and Suse Linux, this security feature is enabled by default. SELinux helps to keep the host secure from certain types of malicious attacks. However, the default settings have been known in many cases to prevent Java from running properly.

To fix this issue, choose one of the following options:
• Configure SELinux so that it knows that the Tivoli Workload Scheduler Java related processes are acceptable to run.
• Change the mode of SELinux to Permissive by entering setenforce 0 on the command line. SELinux will be fully enabled again the next time the system is rebooted or if setenforce 1 is entered on the command line. For the Dynamic Workload Console to function, you must set setenforce 0. For more information about setenforce, see the documentation for your operating system.

On Red Hat Enterprise Linux 6.0 installation fails:
On Red Hat Enterprise Linux 6.0, you launched the installation, but it fails with the following error message written in the twsismp.log file:
AWSJIS138E The specified instance name "instance_name" does not exist on the DB2 server.

Cause and solution
The problem occurs because the db2idefs file cannot be found by the installation program.

To solve the problem, perform the following steps:
1. Add to the PATH environment variable the path where the db2idefs file is located, for example /opt/ibm/db2/V9.7/instance.
2. Launch the installation again.

Problems with the silent installation
The following problem could occur with the silent installation:
• “Silent installation fails without writing a log”

Silent installation fails without writing a log:
You have launched the silent installation but it fails without writing a log.

Cause and solution
The response file is corrupt, or not syntactically correct.
To check the problem, run the setup adding the parameter -is:javaconsole. Correct the syntax of the response file, comparing your version with the supplied templates, or recreate it from the template if it is not readable.

**Problems with installations using the twsinst script**

The following problems could occur:

- An installation with twsinst fails with a return code that does not indicate the reason for failure
- An incorrect password is supplied for the TWS_user on UNIX” on page 200

An installation with twsinst fails with a return code that does not indicate the reason for failure:
If an error occurs during an unattended installation process that makes use of the twsinst script, it can display a return code that is not documented.

**Cause and solution**

Several twsinst error situations give the same return code that is used in the error message that gives the failure. The various error situations have not been documented, because other error messages in the log explain the precise error.

Follow the sequence of installation messages in the log and determine from their context the reason for the problem. Correct the problem and rerun the installation.

**Problems installing the application server**

The following problems might occur:

- The application server profile creation fails
- The application server installation fails on a Windows 2003 domain with an apparent credentials problem but the credentials are correct” on page 206

The application server profile creation fails:
The installation stops in the step "Install with rollback the Tivoli Workload Scheduler modelling and planning server, version 8.5” because it cannot create the application server profile.

When you check the application server trace file

$WAS_HOME/profiles/TIPProfile.deleted/logs/wsadmin.traceout

you find the following line (it is shown here split into three lines):

[1/17/06 17:16:46:886 CST] 0000000a WorkSpaceMast E
WKSP0020E: Error getting meta data repository root
$WAS_HOME/eWAS/profiles/TIPProfile/config/./repository

The variable $WAS_HOME is the directory where the application server is installed.

**Cause and solution**

This error indicates that the folder repository is missing in the installation of the embedded WebSphere Application Server. This means that the following folder is missing or damaged in the path where you placed the installation images:

$PLATFORM_IMAGES_ROOT/EmbeddedExpress/profileTemplates/default/documents/config/./repository
The variable \$PLATFORM\_IMAGES\_ROOT is the location of images for the selected platform, for example, Solaris.

Compare the corresponding files on the distribution media and the location where you copied the installation images.
- If the files are different, the copy of the installation images from the distribution media to the location from which you are using them did not complete correctly. Ensure there is sufficient disk space. Ensure you are using the binary option if using ftp. Recopy the files and rerun the Tivoli Workload Scheduler component installation, or rerun the installation directly from the distribution media.
- If the files are in the same correct path (as indicated above) and are the same, there might be an internal error; contact IBM Software Support for assistance.

**The application server installation fails on a Windows 2003 domain with an apparent credentials problem but the credentials are correct:**
You are installing on a Windows 2003 domain any Tivoli Workload Scheduler component that also installs the application server, but the installation fails at the step where the application server is being installed. The error message given is:

```java
com.ibm.websphere.security.auth.WSLoginFailedException:
Authentication failed for user mdm84 with the following error message
Logon failure: unknown user name or bad password.
```

When you check the credentials you find that the User ID and password are correct.

**Cause and solution**

See “On a Windows 2003 domain, the application server installation fails with an apparent credentials problem but the credentials are correct” on page 196 for the cause and solution.

**Other installation problems**
The following miscellaneous problems might occur:

- “An installation fails on a UNC mapped drive” on page 207
- “Message "Error writing file = ” received” on page 207
- “Message "Error writing file = 28" received” on page 208
- “Message AWSFAB037E is received on UNIX” on page 208
- “A master domain manager installation fails with message AWSJIS038E” on page 208
- “A dynamic domain manager installation fails with message AWSJIS038E” on page 209
- “A dynamic domain manager or backup dynamic domain manager installation fails after the database configuration” on page 209
- “An installation fails with a problem with the installation images on an NFS mount” on page 210
- “An InstallShield wizard installation fails on AIX or HP-UX with a "run error"” on page 210
- “An InstallShield wizard "Add feature" installation fails” on page 211
- “A software package block installation fails with the message: DISSE0324E” on page 211
- “A software package block installation fails to complete successfully” on page 212
- “The installation fails with the error AWSFAB035E” on page 214
An installation fails on a UNC mapped drive:
You are running an installation with the installation images on a drive mapped using the Universal Naming Convention (UNC). The wizard fails at the first step.

Cause and solution

The Tivoli Workload Scheduler installation wizard methodology does not support UNC mapped drives. Rerun the installation from a drive that is not UNC mapped.

Message "Error writing file = " received:
When performing any type of installation on any operating system, you might receive the following error:

Error writing file = There may not be enough temporary disk space.
Try using -is:tempdir to use a temporary directory on a partition with more disk space.

Note in particular the absence of an error code, which differentiates this message from a very similar message, with error code 28, that indicates that you are not logged on as root (see "Message "Error writing file = 28" received" on page 208).

Cause and solution

Normally this error means what it says; the solution is as follows.

First, try to redirect the installation to use a different temporary directory, by adding the -is:tempdir.<temp_dir_path> variable to the installation command.

If this does not work, you must use one of these two methods to give more space to the swdis directory:
• Either:
  Create a new version in a different file system. The procedure is as follows:
  1. Delete or rename both the work and the backup subdirectories and recreate the directories in a file system with more space in it.
  2. Link the new directories to the .swdis directory using the ln -s command.
• Or:
  Create a new backup directory in a file system with more space in it, and modify the /etc/Tivoli/swdis.ini file to point to it.

Ensure to modify the correct section of the swdis.ini file, as follows:
  – If you are making a local silent InstallShield wizard installation that uses the disconnected command line (wdinstsp), modify the value of the backup_dir key in the [#MOBILE] section.
  – If you are making a remote installation using Tivoli Configuration Manager, you must identify the section relative to the endpoint chosen as the target (for example, [lab133080_aix]), and modify the backup_dir key in that section.
**Message "Error writing file = 28" received:**
When performing any type of installation on any operating system, you might receive the following error:

```
Error writing file = 28 There may not be enough temporary disk space. 
Try using -is:tempdir to use a temporary directory on a partition 
with more disk space.
```

Note in particular the error code 28, which differentiates this message from a very similar message, without error code 28, that does indicate disk space problems (see Message "Error writing file = 28" received " on page 207).

**Cause and solution**

This error does not mean exactly what it says. When performing a silent installation of a fix pack on UNIX, and possibly in certain other circumstances, this error message might mean that you are not logged on as root.

Make sure that you are logged onto the workstation as root before running the silent installation:

```
/SETUP.bin -options <path_to_patchInstall.txt> -silent
```

**Message AWSFAB037E is received on UNIX:**
The *twinst* script installation fails on UNIX with the following error message:

```
AWSFAB037E The twinst script is being run from the wrong directory. 
AWSFAB038I Mount the TWS installation CD and run the twinst utility 
placed there.
```

**Cause and solution**

These error messages are received when attempting to install Tivoli Workload Scheduler on a UNIX operating system using the *twinst* utility copied from the installation DVD to the home directory of the user that is nominated as the TWS_user during the installation. The installation fails and no log files are generated.

You can run *twinst* from the following places:

- The Tivoli Workload Scheduler DVD
- A disk image of the DVD
- A copy of the *twinst* utility and its associated files placed in any local directory other than the home directory of the user that is going to be nominated as the TWS_user during the installation.

**A master domain manager installation fails with message AWSJIS038E:**
You are installing a domain manger and the installation fails with the following error message:

```
AWSJIS038E An internal error has occurred. 
An unspecified internal error has occurred during the installation process.
```

moreover, the output of the installation step list if you are performing an interactive installation, or the \inst_logfile_dir\itemConfigure_TWS_DBOut.xml files if you are performing a silent installation, reports the following information:

```
Database DDMSM already present 
Checking if DDMSM is a TWS database 
DDMSM is not a TWS database.
```
Where inst_logfile_dir is the directory where all the installation log files are saved.

**Cause and solution**

You are installing a master domain manager and the installation fails because you specified in the **Database name** field the name of a database already used by a dynamic domain manager. Run the installation again providing a different name for the database.

**A dynamic domain manager installation fails with message AWSJIS038E:**
You are installing a dynamic domain manager and the installation fails with the following error message:
AWSJIS038E An internal error has occurred.
An unspecified internal error has occurred during the installation process.

moreover, the output of the installation step list if you are performing an interactive installation, or the \inst_logfile_dir\ItemConfigure_TWS_DBOut.xml file if you are performing a silent installation, reports the following information:
Database TWSBM already present
Checking if TWSBM is not a TWS database
TWSBM is a TWS database. It must be a TDWB database.
One or more errors has occurred during the database setup.

Where inst_logfile_dir is the directory where all the installation log files are saved.

**Cause and solution**

You are installing a dynamic domain manager, the installation fails because you specified in the **Database name** field the name of a database already used by a master domain manager. Run the installation again providing a different name for the database.

**A dynamic domain manager or backup dynamic domain manager installation fails after the database configuration:**
You are installing a dynamic domain manager or a backup dynamic domain manager and the installation fails after the database configuration step.

**Solution**

In this case you have to uninstall the dynamic domain manager or backup dynamic domain manager manually performing the following steps:
1. From the **TWS_home_directory/TDWB/bin** directory, run the following command:
   `exportserverdata -dbUsr <username> -dbPwd <password> [exportFile <export_file>]`

   where:
   - **username**
     Specify the name of the user required to access the Tivoli Workload Scheduler database server.
   - **password**
     Specify the password of the user required to access the Tivoli Workload Scheduler database server
   - **export_file**
     Specify the name of the text file that is created. The default is `server.properties`. 

Part 2. Tivoli Workload Scheduler  209
2. Open the file you created and edit it by removing the entry related to the failed installation. For example:
   http://9.168.117.211:31115/JobManagerRESTWeb/JobScheduler
3. Update the database definition by running the following command:
   importserverdata -dbUsr <username> -dbPwd <password> [importFile <import_file>]

   where <import_file> is the name of the file you created using the exportserverdata command. The default is server.properties.

An installation fails with a problem with the installation images on an NFS mount:
The installation images are on an NFS mount. The installation fails and the log shows messages similar to the following:
cannot start <file_name>
No such file or directory <file_name>"

where <file_name> is a file in the directory structure of the installation images.

Cause and solution
The NFS mount is corrupt. Refresh the NFS mount by issuing unmount and then mount commands. Retry the failed step (or the entire installation, depending on what went wrong and at what point in the installation).

An InstallShield wizard installation fails on AIX or HP-UX with a "run error":
You are trying to install Tivoli Workload Scheduler using the InstallShield wizard on an AIX or HP-UX operating system. The installation fails giving the following exception in thread "main":

```
java.lang.NoClassDefFoundError: run error
```

Cause and solution
This caused by a combination of display, Java, and binary issues.

To resolve this problem, perform the following steps:
1. Ensure Quality Pack 10 or higher is installed.
2. Run xhost + and re-export the display.
3. Retry the installation using the SETUP.bin binary located at the root of the DVD.
   This copies the appropriate media to /tmp and runs the installation from there.

   **Note:** If you have insufficient space in /tmp, run the SETUP.bin command with the -is:tempdir .<temp_dir_path> variable.

An InstallShield wizard installation does not start and displays the message "A supported JVM is missing":
You are installing on UNIX or Linux and the interactive or silent installation does not start. The message "A supported Java Virtual Machine is missing" displays.

Cause and solution
On very slow environments you could experience this installation problem and error message. However, this InstallShield message is incorrect. The problem is that
the InstallShield Java Virtual Machine verification routine times out due to the slow environment on which you are running.

InstallShield has the following parameter in the installation launcher:

-is:jvmtime timeout in seconds

where timeout in seconds is the amount of time InstallShield waits before timing out the installation. If you have a slow environment, set this parameter to a high value to enable the Java Virtual Machine recognition routine enough time to obtain the value before timeout. Initially, set the timeout in seconds parameter to 60, and increment it by 60 until the installation successfully launches.

An InstallShield wizard "Add feature" installation fails:
You are trying to add a feature to an existing Tivoli Workload Scheduler installation, using the Add feature option in the wizard. You have started the wizard from the Tivoli Workload Scheduler DVD, but have previously copied the DVD image of the feature to your hard disk. When the wizard asks you to supply the path to the feature installation, you eject the product DVD before supplying the hard disk path information. The installation fails.

Cause and solution
This is a known problem with the InstallShield wizard. If an installation starts from a DVD, the InstallShield wizard expects to find a DVD in the DVD drive.

To correct this problem, put the product DVD, or any other DVD, back in the DVD drive (the InstallShield wizard requires a DVD, but it can be any DVD).

A software package block installation fails with the message: DISSE0324E:
You launched an installation of either the full product, a component, or a fix-pack, that uses the software package blocks of the software distribution component of IBM Tivoli Configuration Manager. The installation fails, giving the following messages:

```
DISSE0282E Error compressing file <file_name> in the software Package block.
DISSE0324E Cannot create backup package
DISSE0005E Operation unsuccessful
```

Cause and solution
The installation using a software package block is unable to check that there is sufficient space for the backup it performs. The backup requires at least 80 MB of disk space. The directory used for the backup is determined by the parameter backup_dir in the [#MOBILE] section of the swdis.ini file. The following is an example of this section of the file:
If there is insufficient space in this directory, the error messages shown above are displayed.

Because the Tivoli Configuration Manager installation was unable to start, the restore script twsRestore cannot be used. The recovery procedure is as follows:

1. If you are performing a full-product installation, manually delete the following files and directories:
   - twsRestore.sh or twsRestore.cmd (as appropriate)
   - twsRemove.sh or twsRemove.cmd (as appropriate)
   - _uninstall directory

2. Resolve the file space problem, as follows.
   First, try to redirect the installation to use a different temporary directory, by adding the \
   **-is:tempdir .<temp_dir_path>** variable to the installation command.
   If this does not work, you must use one of these two methods to give more space to the swdis directory:
   - Either:
     Create a new version in a different file system. The procedure is as follows:
     a. Delete or rename both the work and the backup subdirectories and recreate the directories in a file system with more space in it.
     b. Link the new directories to the .swdis directory using the **ln -s** command.
   - Or:
     Create a new backup directory in a file system with more space in it, and modify the /etc/Tivoli/swdis.ini file to point to it.
     Ensure to modify the correct section of the swdis.ini file, as follows:
     - If you are making a local silent InstallShield wizard installation that uses the disconnected command line (wdinstsp), modify the value of the backup_dir key in the [#MOBILE] section.
     - If you are making a remote installation using Tivoli Configuration Manager, you must identify the section relative to the endpoint chosen as the target (for example, [lab133080_aix]), and modify the backup_dir key in that section.

3. Run the installation again.

**A software package block installation fails to complete successfully:**
You launched an installation that uses the software package blocks of the software distribution component of IBM Tivoli Configuration Manager. The installation fails.

**Cause and solution**
Problems when installing remotely with a software package block can often be difficult to solve, because it might be more difficult to set the remote environment correctly so that the installation runs successfully. For this reason, the software package block is supplied with a series of keys that switch on or off its installation activities. These action keys are set by default to `true`, so that the installation completes normally. If it fails for a reason that you know you can resolve afterwards, you can retry it, setting one or more of these keys to `false`, so that the installation process does not attempt to perform that or those steps.

For example, if the installation fails when trying to back up the previous installation, and you know that you can proceed without making a backup, you can eliminate this action from the installation by setting its action key to `false`. You re-launch the installation, which does not perform the backup step, but otherwise completes successfully.

A description of the processing carried out in each step is given in the *Tivoli Workload Scheduler: Planning and Installation Guide*.

**Note:** The installation steps are always the same, whatever installation method you use.

The details of the action keys are as follows (set any of them to `false` to not perform that action):

- **execActionTools** = "true"
  This controls all of the other keys. If you set it to `false`, none of the other actions take place (their settings are ignored).

- **execTwsStopAction** = "true"
  This controls whether or not the installation stops existing Tivoli Workload Scheduler processes on the target workstation.

- **execTwsCleanAction** = "true"
  This controls whether or not the installation cleans up an existing installation before upgrading it.

- **execTwsUndoAction** = "true"
  This controls whether or not the undoable installation script is run.

- **execTwsBackupAction** = "$\text{backup}\)"
  This controls whether or not the installation takes a backup of an existing installation before commencing the installation. By default this value is set to "false".

- **execTwsUserAction** = "true"
  This controls whether or not the installation creates or modifies the `<TWS_user>` details.

- **execTwsConfigAction** = "true"
  This controls whether or not the installation configures Tivoli Workload Scheduler after the installation.

- **execTwsStartUpAction** = "true"
  This controls whether or not the installation starts up Tivoli Workload Scheduler after the installation.

- **execTwsCommitAction** = "true"
  This controls whether or not the installation issues a software distribution `commit` action to complete the installation (see the Tivoli Configuration Manager publications for more details).
When you have resolved the problem, reset these keys to true for any other installation using the software package block.

See the Tivoli Configuration Manager publications for details of how to change the values of a parameter in a software package block.

**The installation fails with the error AWSFAB035E:**
You are trying to install Tivoli Workload Scheduler or one of its fix packs. The installation fails with the following error:

```
AWSFAB035E The installation has failed. For more details see the log file: /tmp/tws84/summary.log.
```

The installation log contains the Software Distribution error message DISSE0006E:

```
Explanation: The operation cannot be completed because of an internal error (for example, a memory allocation failure)
System Action: Operation failed.
```

The log also contains details of the last internal command it ran:

```
+ wdinstsp -f -D promote=false -D upgrade=false -D fresh_install=true
```

**Cause and solution**

This is a problem related to incompatibility between the Tivoli Workload Scheduler installer and the current version of IBM Tivoli Configuration Manager. Follow these steps:

1. Check the version of Tivoli Configuration Manager that is installed, using the command `wlsinst -ah`. Check if any patches are installed.
2. If you are using Tivoli Configuration Manager version 4.2, with fix pack 4.2-SWDDGW-F1P1 (or later for the same component), there is a compatibility problem because the installation of Tivoli Workload Scheduler version 8.4 and its fix packs are only compatible with the GA version of the Tivoli Configuration Manager SWDDGW component.
   In this case you must either choose a different installation method that does not use Tivoli Configuration Manager or uninstall the Tivoli Configuration Manager 4.2-SWDDGW fix pack or packs until the installation is complete.
3. The Tivoli Workload Scheduler GA and fix pack installation uses the Tivoli Configuration Manager version 4.2 disconnected command line. A problem has been discovered with this component which requires you to do the following:
   a. Install Tivoli Configuration Manager fix pack 4.2-TCM-FP02 on the workstation.
   b. Run the `wconvcat` command (described in the 4.2-TCM-FP02.README file) to restore the functionality of the disconnected command line.
   c. Retry the failed installation.

**An agent installation fails if you launch it from a shell where you set the environment variables using the swd_env script:**
The following error occurs when you install an agent:

```
AWSFAB035E The installation has failed.
```

**Cause and solution**
You can receive this error if you are running the installation form a shell script where you previously set up the Software Distribution environment variables using the `swd_env.cmd` or the `./swd_env.sh`.

Resume the installation from a shell script where you did not set the environment variables and complete the installation.

**The installation fails with the error AWSGAB566E:**

You are trying to install Tivoli Workload Scheduler, and the installation fails with the following error:

```
AWSGAB566E There is not enough disk space available in the following supplied directory to complete the installation: <directory_name>. The installation requires <required_space> megabytes, but only <available_space> megabytes are available.
```

**Cause and solution**

The probable cause of this error is that either the file set where the product is to be installed, or the file set where the temporary installation files are being written, is not large enough. Information about the disk space requirements is given in the Tivoli Workload Scheduler System Requirements Document at [http://www.ibm.com/support/docview.wss?rs=672&uid=swg27019747](http://www.ibm.com/support/docview.wss?rs=672&uid=swg27019747)

However, this error is also given if the virtual memory file (sometimes called page file or swap space) on your hard disk is not large enough.

Thus, if there seems to be sufficient disk space, check also the virtual memory that you have allocated to the hard disks. The installation requires at least 256 MB of virtual memory on any operating system.

**The commit step fails:**

You are trying to install Tivoli Workload Scheduler, and the commit step fails

**Cause and solution**

If the mapped attribute, LDAPUSERIdMap is different from the login attribute, LDAPUserFilter, the commit step fails. You must dump the security file and insert the mapped value (see error message). Then, manually perform the failed composer command (see error message). Finally, set the failed step as successful and resume the installation.

**Uninstalling and reinstalling a dynamic agent with the same name on the same workstation:**

If you have to uninstall a dynamic agent from a workstation and want to reinstall it with same name on the same workstation, perform the following steps to override the existing agent definition:

1. Obtain the ID of the dynamic agent running the following command: `conman cpupinfo dynamic attitude on the workstation in the plan.`

2. Add the ID of the dynamic agent in the jobmanager.ini file using the `AgentID` keyword, as shown in the following example:

```ini
[ResourceAdvisorAgent]
FullyQualifiedHostname = geneva12.paralab.it.fsx.com
ResourceAdvisorUrl = https://localhost.localdomain:31116/JobManagerRESTWeb
/JobScheduler/resourceCPUScannerPeriodSeconds = 10
ScannerPeriodSeconds = 120
```
Miscellaneous failures:
The installation fails and the cause is not immediately obvious from the log messages.

Cause and solution

The cause of the failure could be any of the following:

The FTP transfer of the files to the node was not done in binary mode
You copied the install directory from the DVD to the local hard disk using FTP, but did not specify the binary option. Make sure the entire directory is transferred by FTP in binary mode.

Note: The directory on the local hard disk can have any name, but it is important to have a parent directory available for the twsinst installation, because some temporary files need to be located there.

For example:
/temp/HP-UX
or
/temp/TWS84/HP-UX

There is not enough disk space available for the installation
Check that there is enough disk space for the installation on your chosen fileset.

See the Tivoli Workload Scheduler System Requirements Document at http://www.ibm.com/support/docview.wss?rs=672&uid=swg27019747 for more information about the amount of space necessary for installation.

File names did not retain their original case
On UNIX, check that file names retain their case. For example, the file "TWS_size.txt" cannot be "tws_size.txt".

One or more required files were not copied from the root of the installation

DVD Check that the number of files copied from the DVD is the same as that on the DVD. If not, copy the files again.

You launched a second installation before the first one had successfully finished.

If you launch an installation (of an additional component, for example) while another installation is still running, both installations might fail, or the second installation might try and resume the first, as if you had terminated the first installation part way through and now want to continue.

Depending on the stage that the first installation reached, you might just be able to close the second installation and let the first one finish.
However, if one or both have failed, you might need to uninstall and then start the installation again.
The InstallShield wizard installation on Oracle fails with the error AWSJIS145E:
You are installing on Oracle using the InstallShield wizard. The wizard displays the error message:
AWSJIS145E "The supplied credentials for user Tivoli Workload Scheduler Oracle Database user are not correct"

Cause and solution
If you open the installation log file twsismp.log, you can see the error:

```
DBCfgInfo, dbg, validate_twsDBUserAuthInfo: result of the query for the existence of the TWS Oracle User is:
ERROR:ORA-28002: the password will expire within 7 days 0
ORA-01017: invalid username/password; logon denied
```

The problem is that the password of the Oracle system account, used during the installation, is about to expire. Change the password of the Oracle system account and restart the wizard installation.

The InstallShield wizard installation of backup master domain manager does not correctly manage the localopts file:
You installed a backup master domain manager using the InstallShield wizard. The installation completes successfully, but the localopts file does not contain the following sections:

```
#----------------------------------------------------------------------------
# Attributes for CLI connections
#
# Master hostname used when attempting a connection.
HOST = 127.0.0.1
# Protocol used to establish a connection with the Master.
PROTOCOL = https
PORT = 31116 # Protocol port
#PROXY = proxy.userlocal
#PROXYPORT = 3333
TIMEOUT = 3600 # Timeout in seconds to wait a server response
#CLISSLSERVERAUTH = yes
#CLISSLCIPHER = MD5
#CLISSLSERVERCERTIFICATE =
#CLISSLTRUSTEDDIR =
DEFAULTTWS = NC112015
USEROPTS = useropts_mdm86
#----------------------------------------------------------------------------

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

Cause and solution
If you perform some next and back actions in the wizard panels, the installation process cannot create the CLI connections and Event Management parameters sections of the localopts file. After installation, manually add the sections to the file.

The installation of additional plug-in by does not have enough temporary space:
The installation of an additional plug-in using the Tivoli Workload Scheduler for Additional Plug-ins fails with the message:

```
WARNING:/tmp does not have enough disk space!
    Attempting to use / for install base and tmp dir
```
Cause and solution

If the temporary directory does not have enough space, redirect the installation process to another temporary directory, set the InstallAnywhere variable IATEMPDIR:

Windows operating systems
1. set IATEMPDIR=<new_temp_dir>
2. Start the installation.

UNIX operating systems
1. export IATEMPDIR=<new_temp_dir>
2. Start the installation.

Upgrade problems
The following problems could be encountered.
- "Variables not resolved after upgrade"
- "Default variable table not accessible after upgrade"
- "Registry file information not found during upgrade" on page 230

Variables not resolved after upgrade:
After upgrading to version 8.5, global variables are not resolved.

Cause and solution

During the upgrade to version 8.5, all the security file statements relating to your global variables were copied by the install wizard into a default variable table in the new security file. Global variables are disabled in version 8.5, and can only be used through the variable tables. If you subsequently rebuilt the security file using the output from your previous dumpsec as input to the new makesec, you will have overwritten the security statements relating to your default variable table, so no user has access to the default variable table.

If you have a backup of your security file from prior to when you ran makesec, run dumpsec from that, and merge your old dumpsec output file with your new one, as described in the upgrade procedure in the Tivoli Workload Scheduler: Planning and Installation Guide.

If you do not have a backup, create the default variable table security statement, following the instructions about configuring the security file in the Tivoli Workload Scheduler: Administration Guide.

Default variable table not accessible after upgrade:
After upgrading to version 8.5, your default variable table is not accessible by any user.

Cause and solution

This problem has exactly the same cause and solution as the preceding - see "Variables not resolved after upgrade."

Installing:
This chapter describes how to install the Dynamic Workload Console. It is divided into the following sections:
Installing the Dynamic Workload Console:
This section explains how to install the Dynamic Workload Console using the available installation methods. It is divided into the following topics:

- Using the launchpad
- Using the installation wizard

Using the launchpad:
You can install the Dynamic Workload Console using the launchpad. Use the instructions for launching and running the launchpad at Installing from the launchpad and choose the Dynamic Workload Console installation option in the launchpad. Follow the on-screen instructions. The launchpad runs the installation wizard with some of the options pre-filled. Follow the instructions for Using the installation wizard to complete the process.

Using the installation wizard:
Follow these steps to install the Dynamic Workload Console using the installation wizard:

1. Browse to the setup directory and start the installation by running the setup file. The installation wizard first checks if there is enough free space available in the Java temporary directory. If not, the installation exits, and you must increase the size of the Java temporary directory, as described in the Tivoli Workload Scheduler System Requirements Document at http://www.ibm.com/support/docview.wss?rs=672&uid=swg27019747, before rerunning the installation wizard.

2. Select the language to use while installing the Dynamic Workload Console, and click OK.

3. In the welcome panel, click Next to continue with the installation.

4. Read and accept the license agreement. Click Next.

5. Select the Tivoli Integrated Portal instance. Choose among the following:
   - If you choose to install a new instance of the Tivoli Integrated Portal or if you choose to install on an existing instance of Tivoli Workload Automation that does not contain the embedded WebSphere Application Server, perform the steps in Installing a new instance of the Tivoli Integrated Portal on page 220. Perform this installation if you do not have the Tivoli Integrated Portal already installed or if you have installed a Tivoli Workload Scheduler component that does not install the embedded WebSphere Application Server for example a fault-tolerant agent.
   - If you choose to install on an existing Tivoli Workload Automation instance that contains the embedded WebSphere Application Server, perform the steps in Installing on an existing instance of the embedded WebSphere Application Server on page 222. Perform this installation if you have...
already installed a Tivoli Workload Automation component that installs the embedded WebSphere Application Server also, for example the master domain manager.

- If you choose to install on top of your existing instance of Tivoli Integrated Portal, perform the steps in “Installing on your existing instance of Tivoli Integrated Portal” on page 223. If you perform this installation if you have already installed a Tivoli Integrated Portal with another Tivoli product. For a list of supported Tivoli Integrated Portals, see the Tivoli Workload Scheduler System Requirements Document at http://www.ibm.com/support/docview.wss?rs=672&uid=swg27019747.

- Starting from V8.6 we do not support anymore the Dynamic Workload Console installed on external WebSphere Application Server. If you do not have the Tivoli Integrated Portal installed, you can install it using the installation DVD or the appropriate eImages as described in the “Installing the Tivoli Integrated Portal on an external WebSphere Application Server from the images” on page 224.

6. Select an installation location. Click Next.

7. Specify the user name and password of the Tivoli Integrated Portal user that you want to use as the Dynamic Workload Console administrator.

   **Note:** The user name and password must be operating systems credentials. On Windows operating systems, if the user name you specify does not exist, a new operating system user is created.

   The user name must be unique, 3 to 60 characters in length, and contain only the characters a-z, A-Z, 0-9, period (.), hyphen (-), underscore (_), and double-byte character set (DBCS) characters.

   The password must be 5 to 16 characters in length and contain only the characters a-z, A-Z, 0-9, period (.), hyphen (-), and underscore (_).

   The user must be created as described in Rules for using a Federated User Registry with Tivoli Workload Scheduler.

   Confirm the password and click Next.

8. Choose a new path to install into or choose the path of the existing Tivoli Workload Automation instance. Choose the path where you want to install, from now on referred to as `twa_install_dir`, or accept the default path, and click Next.

   Make sure that the installation path is 32 characters or less in length and that it does not contain special characters.

   **Installing a new instance of the Tivoli Integrated Portal:**
   The following applies if you are installing a new Tivoli Workload Automation instance or if you are installing over an existing Tivoli Workload Automation instance where the embedded WebSphere Application Server has not yet been installed. You are in this case if you do not have the Tivoli Integrated Portal already installed or if you have installed a Tivoli Workload Scheduler component that does not install the embedded WebSphere Application Server for example a fault-tolerant agent. In this case Tivoli Workload Scheduler installs the embedded WebSphere Application Server and the Tivoli Integrated Portal.

   Follow these steps if you selected to install the Tivoli Integrated Portal and the Dynamic Workload Console:

   In the installation choice window, select one of the following installation types.
Default Installation
If you want to use the default Tivoli Integrated Portal settings, proceed with the installation as described in “Default installation.”

Advanced Installation
If you want to customize the Tivoli Integrated Portal settings, proceed with the installation as described in “Advanced installation.”

Default installation:
Follow these steps to proceed with a default installation:
1. To start the installation, check that the values displayed in the installation summary window are correct and click Install.
2. When the installation completes successfully, a window opens showing links to the user interface on the Tivoli Integrated Portal. For more information, see “Accessing the Dynamic Workload Console” on page 226. If the installation fails, the window contains the list of the items that were not installed and the location of the log file. Click Finish.

Advanced installation:
Perform the following steps to proceed with an advanced installation:
1. Specify the following port numbers for the Tivoli Integrated Portal or accept the default values. These are embedded WebSphere Application Server ports used by Tivoli Integrated Portal.

   HTTP transport
   The number of the port that the portal uses for HTTP transport. The default value is 29080.

   HTTPS transport
   The number of the port that the portal uses for secure HTTP transport (HTTPS). The default value is 29443.

   Bootstrap
   The port number for the bootstrap function. The default value is 22809.

   SOAP connector
   The port number for the Simple Object Access Protocol (SOAP) connector on the portal. The default value is 28880.

   SAS server authentication listener
   The SAS SSL server authentication listener port number on the portal. The default value is 29401.

   CSIv2 server authentication listener
   The CSIv2 SSL ServerAuth Listener port number on the portal. The default value is 29403.

   CSIv2 Client Authentication Listener
   The CSIv2 SSL ClientAuth Listener port number on the portal. The default value is 29402.

   ORB listener
   The ORB listener port number on the portal. The default value is 29100.

   Administrative console
   The HTTP administrative console port on the portal. The default value is 29060.

   Administrative console secure
   The HTTP administrative console secure port on the portal. The default value is 29043.
Install the Dynamic Workload Console on an existing instance of the embedded WebSphere Application Server:

1. Select the existing Tivoli Workload Automation directory.
2. Supply the username and password of the existing instance of the embedded WebSphere Application Server.

Note: If you have already installed WebSphere Application Server into your existing Tivoli Workload Automation instance but do not know the username, click Retrieve. The username is retrieved but you still must provide the password. This operation may take a few minutes. If you are performing a silent installation, to find these credentials, run showSecurityProperties before running the installation.

3. Select if you want the administrator to access the Tivoli Workload Scheduler console, the Dynamic Workload Broker console, or both. Click Next.

Note: If you select one of the two available user interfaces, after installing you can authorize the user to access the other user interface by assigning him one of the predefined roles created by the installation process. For more information, see the information about configuring the Dynamic Workload Console in the Tivoli Workload Scheduler: Administration Guide.

4. To start the installation, check that the values displayed in the installation summary window are correct and click Install.

Specify the following port numbers for the Tivoli Integrated Portal or accept the default values. These are embedded WebSphere Application Server ports used by Tivoli Integrated Portal

**IPC connector**
The IPC connector on the portal. The default value is **29314**.

**REST notification**
The REST notification port on the portal. The default value is **29324**.

**DCS Unicast port**
The DCS Unicast port on the portal. The default value is **29353**.

Click Next.

2. Complete the installation by following the steps described in “Default installation” on page 221.

Installing on an existing instance of the embedded WebSphere Application Server:

You perform this installation if you have already installed a Tivoli Workload Automation component that installs the embedded WebSphere Application Server also, for example the master domain manager. Follow these steps to install the Dynamic Workload Console on an existing instance of the embedded WebSphere Application Server:

1. Select the existing Tivoli Workload Automation directory.
2. Supply the username and password of the existing instance of the embedded WebSphere Application Server.

Note: If you have already installed WebSphere Application Server into your existing Tivoli Workload Automation instance but do not know the username, click Retrieve. The username is retrieved but you still must provide the password. This operation may take a few minutes. If you are performing a silent installation, to find these credentials, run showSecurityProperties before running the installation.

3. Select if you want the administrator to access the Tivoli Workload Scheduler console, the Dynamic Workload Broker console, or both. Click Next.

Note: If you select one of the two available user interfaces, after installing you can authorize the user to access the other user interface by assigning him one of the predefined roles created by the installation process. For more information, see the information about configuring the Dynamic Workload Console in the Tivoli Workload Scheduler: Administration Guide.

4. To start the installation, check that the values displayed in the installation summary window are correct and click Install.

Specify the following port numbers for the Tivoli Integrated Portal or accept the default values. These are embedded WebSphere Application Server ports used by Tivoli Integrated Portal

**IPC connector**
The IPC connector on the portal. The default value is **29314**.

**REST notification**
The REST notification port on the portal. The default value is **29324**.

**DCS Unicast port**
The DCS Unicast port on the portal. The default value is **29353**.

5. When the installation completes successfully, a window opens showing links to the user interface on the Tivoli Integrated Portal. For more information, see “Accessing the Dynamic Workload Console” on page 226. If the installation fails, the window contains a list of the items that were not installed and the location of the log file. Click Finish.
Installing on your existing instance of Tivoli Integrated Portal:

You perform this installation if you have already installed a Tivoli Integrated Portal with another Tivoli product. Follow these steps to install the Dynamic Workload Console on top of an existing Tivoli Integrated Portal instance:

1. Select the existing Tivoli Integrated Portal instance over which you want to install the Dynamic Workload Console by specifying the installation path.
2. Specify the user ID and password of an existing Tivoli Integrated Portal user that you want to set as the Dynamic Workload Console administrator.

**Note:** If you select one of the two available user interfaces, after installing you can authorize the user to access the other user interface by assigning him one of the predefined roles created by the installation process. For more information, see the information about configuring the Dynamic Workload Console in the *Tivoli Workload Scheduler: Administration Guide*.

3. To start the installation, check that the values displayed in the installation summary window are correct and click Install.
4. When the installation completes successfully, a window opens showing links to the user interface on the Tivoli Integrated Portal. For more information, see “Accessing the Dynamic Workload Console” on page 226. If the installation fails, the window contains a list of the items that were not installed and the location of the log file. Click Finish.

Performing a silent installation:

You can run the installation in unattended mode from the command line by adding the **silent** parameter when running the setup installation file. Perform the following steps:

- Run the installation as root on UNIX operating systems, or as Administrator on Windows operating systems.
- Specify all the settings that are prompted when installing using the installation wizard.

The installation settings are provided using a response file.

Edit the response file templates provided on the installation DVDs in the \tdw\responsefiles\ directory. Instructions for customizing the files are included in the files as commented text. For details about response file properties, see “The Dynamic Workload Console response file properties” on page 303.

**Table 17** lists the response files and the types of installation each performs by operating system:

<table>
<thead>
<tr>
<th>Type of installation</th>
<th>Response file to use on Unix</th>
<th>Response file to use on Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fresh Dynamic Workload Console on existing TWA instance</td>
<td>TDWC86_FRESH_existTWA_UNIX.txt</td>
<td>TDWC86_FRESH_existTWA_WIN.txt</td>
</tr>
<tr>
<td>Fresh Dynamic Workload Console on an external Tivoli Integrated Portal</td>
<td>TDWC86_FRESH_extTIP_UNIX.txt</td>
<td>TDWC86_FRESH_extTIP_WIN.txt</td>
</tr>
<tr>
<td>Fresh Dynamic Workload Console on a new TWA instance</td>
<td>TDWC86_FRESH_newTWA_UNIX.txt</td>
<td>TDWC86_FRESH_newTWA_WIN.txt</td>
</tr>
<tr>
<td>Uninstall the Dynamic Workload Console</td>
<td>TDWC86_UNINSTALL.txt</td>
<td>TDWC86_UNINSTALL.txt</td>
</tr>
</tbody>
</table>
Table 17. Dynamic Workload Console response files (continued)

<table>
<thead>
<tr>
<th>Type of installation</th>
<th>Response file to use on Unix</th>
<th>Response file to use on Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upgrade the Dynamic Workload Console on existing Tivoli Workload Automation instance (embedded WebSphere Application Server)</td>
<td>TDWC86_UPGRADE_embeddedWAS_UNIX.txt</td>
<td>TDWC86_UPGRADE_embeddedWAS_WIN.txt</td>
</tr>
</tbody>
</table>

**Note:** In the upgrade scenarios, choose the embedded version of IBM Websphere Application Server that you originally chose when you installed the Dynamic Workload Console version 8.4 or higher.

To install in silent mode, perform these steps on the computer on which you want to install the Dynamic Workload Console:

1. Copy the sample response file for that operating system to a local temporary directory.
2. Customize the options contained in the response file to suit your requirements and environment. For information about the available options, see "The Dynamic Workload Console response file properties" on page 303.
3. Run the following command:
   - **Windows operating system:**
     ```
     SETUP.exe -options response_file.txt -silent
     ```
   - **UNIX and Linux operating systems:**
     ```
     ./SETUP.bin -options response_file.txt -silent
     ```
     where `response_file` is the full path name.
4. Check the result of the silent installation as follows:
   - **Windows operating system:**
     The installation command is asynchronous, meaning that when it is issued it starts an installation procedure and then ends without returning any value or message. To know whether or not the silent installation ran successfully, see the installation result reported in the `tdwcinstall.log` installation log file stored in the temporary directory.
   - **UNIX and Linux operating systems:**
     The installation command is synchronous and it returns 0 if the installation ran successfully, or a nonzero value if the installation failed.

**Note:** For information about the installation result, see the `tdwcinstall.log` installation log file stored in the temporary directory.

*Installing the Tivoli Integrated Portal on an external WebSphere Application Server from the images:*

The following procedure applies if you do not have the Tivoli Integrated Portal installed but you have the WebSphere Application Server installed. To install the Tivoli Integrated Portal, perform the following steps:

1. From the installation DVD or from the downloaded eImages, go to the `TDW_<operating_system>\TDW\WEBUI\<operating_system>TIP\` directory where the `sample_response.txt` file is located.
2. Follow the instructions provided in the `sample_response.txt` file to customize the properties necessary to perform the installation.
3. Run the following command:

```
install.sh/bat <java_jre_16_home> sample_response.txt
```

where `java_jre_16_home` is the path where the Java V16 is installed.

The Tivoli Integrated Portal installation creates the TIPProfile profile into your existing instance of the WebSphere Application Server. After you installed the Tivoli Integrated Portal, you can install the Dynamic Workload Console on this new Tivoli Integrated Portal instance by following the instructions provided in "Installing the Dynamic Workload Console" on page 219.

**Post-installation steps to connect to Tivoli Workload Scheduler Version 8.3 Fix Pack 3:**

To access a Tivoli Workload Scheduler Version 8.3 Fix Pack 3 environment, you must enable Tivoli Workload Scheduler to work with the Dynamic Workload Console.

**Note:**

1. These steps are not necessary to connect to a Tivoli Workload Scheduler environment for any version higher than V8.3 Fix Pack 3. Any upgrades performed after version 8.3 Fix Pack 3 will maintain any changes made during this procedure.

2. If you plan to communicate from the Dynamic Workload Console version 8.4 or higher to Tivoli Workload Scheduler, Version 8.3 Fix Pack 3, make sure that the APAR PK47309 is installed on top of the Tivoli Workload Scheduler engine. For more information, contact IBM Software Support.

3. Before proceeding, it is recommended that you run the `backupConfig.sh` or `backupConfig.cmd` script to backup the Tivoli Workload Scheduler configuration. For information about how to run these scripts, see the Tivoli Workload Scheduler: Administration Guide.

This task must be run on the system where the Tivoli Workload Scheduler engine that you want to connect to is installed:

**Tivoli Workload Scheduler distributed environment**

- On the master domain manager.
- On a full status fault-tolerant agents (FTA) workstation where the Tivoli Workload Scheduler connector is installed.

**Tivoli Workload Scheduler z/OS environment**

On the distributed system where the Tivoli Workload Scheduler z/OS Connector is installed.

Perform the following steps:

1. Make sure that the embedded or external version of WebSphere Application Server, as appropriate, is started on the Tivoli Workload Scheduler workstation and then run the following script:

   **On Windows operating system:**
   
   As Administrator, from the directory `TWS_home\wastools`:
   
   ```
   webui -operation enable -user TWS_user -password TWS_user_pw
   -port TWS_port [-server TWS_server]
   ```

   **On UNIX and Linux systems**
   
   As root, from the directory `TWS_home/wastools`:
   
   ```
   ./webui.sh -operation enable -user TWS_user -password TWS_user_pw
   -port TWS_port [-server TWS_server]
   ```
where:

TWS_user
The Tivoli Workload Scheduler administrator user ID.

TWS_user_pw
The Tivoli Workload Scheduler administrator password.

TWS_port
The SOAP port of the WebSphere Application Server where the Tivoli Workload Scheduler is installed. This is a mandatory setting when using the enable flag. Its default values are 31118 for distributed environments, and 31128 for z/OS environments.

TWS_server
The name of the server specified in the WebSphere Application Server profile used by Tivoli Workload Scheduler. By default the value assigned to this field is server1.

2. Stop and start the external or embedded WebSphere Application Server on the Tivoli Workload Scheduler system where you run the script.

When you have completed these steps, you are ready to create engine connections for the Tivoli Workload Scheduler workstation and to manage your Tivoli Workload Scheduler production environment. For information about how to accomplish these tasks, access the Dynamic Workload Console online general help.

Post-installation steps to configure the use of Lightweight Third-Party Authentication (LDAP):
If the Dynamic Workload Console and the Tivoli Workload Scheduler engine or the Tivoli Workload Scheduler z/OS Connector have been configured with the same LDAP user registry, or are installed on the same computer, you might receive a connection failure. If this happens, use the same Lightweight Third-Party Authentication (LTPA) keys on all servers: the Dynamic Workload Console, the Tivoli Workload Scheduler engine server, and the Tivoli Workload Scheduler z/OS Connector server.

To align the LTPA keys, see the section on configuring the use of Lightweight Third-Party Authentication in the Administration Guide.

Accessing the Dynamic Workload Console:
When the installation of the Dynamic Workload Console completes successfully, a message with links to the Integrated Solutions Console portal is displayed. If you used the silent installation, this information is stored in the tdwinstall.log installation log file. For more details about where to find the installation logs, see “Interactive wizard installation and uninstallation log files” on page 248.

From a supported browser, access one of the following links provided by the installation program:

http://dynamic_workload_console_system:http_port/ibm/console

https://dynamic_workload_console_system:https_port/ibm/console

where:

dynamic_workload_console_system
The hostname or IP address of the system where you installed the Dynamic Workload Console.
**http_port**
The port number used to access the Dynamic Workload Console using an unsecure connection over HTTP. The default value for this port number is 29080 if you installed the Dynamic Workload Console in a new Tivoli Workload Automation instance. If you installed the Dynamic Workload Console into an existing Tivoli Workload Automation instance, the value for this port is inherited. If the existing Tivoli Workload Automation instance contains the current version of Tivoli Workload Scheduler using default ports, the value is **31123**.

**https_port**
The port number used to access the Dynamic Workload Console using a secure connection over HTTPS. The default value for this port number is 29443 if you installed the Dynamic Workload Console as a new Tivoli Workload Automation instance. If you installed the Dynamic Workload Console into an existing Tivoli Workload Automation instance, the value for this port is inherited. If the existing Tivoli Workload Automation instance contains the current version of Tivoli Workload Scheduler using default ports, the value is **31124**.

When connecting to the Tivoli Integrated Portal using an HTTPS connection, if you receive a security alert, proceed with the Dynamic Workload Console working session. If you receive security information windows while navigating through the Tivoli Integrated Portal, choose to display nonsecure items to proceed. If you are using Internet Explorer, you can prevent these windows from opening by setting **Display mixed content** to **Enable** in the Security settings.

In the Tivoli Integrated Portal login portlet, enter the user ID and password you specified during the installation, and click **Log in**.

On the navigation bar on the left, expand the **Tivoli Workload Scheduler** entry to access the Dynamic Workload Console and then the Tivoli Workload Scheduler components. Expand the **Dynamic Workload Broker** entry to access Dynamic Workload Broker environments.

To effectively use the functions of these two products, you must define connections to the Tivoli Workload Scheduler engines and the Dynamic Workload Broker servers.

Without defining engine connections, you can perform only this limited set of operations:

**On Tivoli Workload Scheduler**
- Create browse tasks
- Create report tasks
- Create event management tasks
- Define user preferences

**On Dynamic Workload Broker**
- Define user preferences

If the user ID you used to connect to the Dynamic Workload Console has been assigned a role different from **TWSWEBUIAdministrator** and **TDWBAadministrator**, you will see a subset of the available panels. This subset depends on the authorizations assigned to the role associated to your user ID. For more information about roles, see the information about configuring the Dynamic Workload Console in the **Tivoli Workload Scheduler: Administration Guide**.
If the user ID you used to connect to the Dynamic Workload Console has no role assigned, you do not see the entries for Tivoli Workload Scheduler and Dynamic Workload Broker in the Tivoli Integrated Portal portal navigation tree.

Quick steps to define a Tivoli Workload Scheduler engine connection:
After logging in to the Dynamic Workload Console using the administrator userid or another userid with assigned **TWSWEBUIAdministrator** or **TWSWEBUIConfigurator** roles, use the following steps to create an engine connection to one of your supported Tivoli Workload Scheduler engines.

**Note:** If you installed the Dynamic Workload Console into a Tivoli Workload Automation instance that had the embedded WebSphere Application Server already installed, the connection to the Tivoli Workload Scheduler component (for example, master domain manager, backup master domain manager, or connector) is automatically defined with blank credentials. The connection is shared with all the Dynamic Workload Console users and no further credentials are needed because Single Sign On is automatically implemented for the component. The same situation applies if you install a Tivoli Workload Scheduler component into a Tivoli Workload Automation instance where the Dynamic Workload Console and the embedded WebSphere Application Server are already installed.

1. To expand the tree, click the Dynamic Workload Console and **Tivoli Workload Scheduler**.
2. Select **Quick start**
3. Click **New Engine**.
4. In the Engine Connection Properties window, assign a name to the engine connection and specify:

   **Engine Type**
   Either z/OS or Distributed. This is the type of the Tivoli Workload Scheduler engine to connect to.

   **Hostname**
   The hostname or IP address of system where the distributed engine or the z/OS connector, for z/OS engine types, runs.

   **PortNumber**
   The bootstrap port number for the Tivoli Workload Scheduler engine. Default values are **31117** for distributed engine, and **31217** for z/OS connector.

   **Remote Server Name**
   This setting is valid and mandatory only for z/OS engines. It is the value specified when the engine was created in the z/OS connector. It must exactly match the z/OS connector engine name and is case sensitive. If the engine was defined using the WASTOOLS "createZosEngine" COMMAND, this is the value specified in the **-name** parameter. This is the name of the remote server of the engine as it is specified in the z/OS connector.

   **Userid and Password**
   The user ID and password that are used to connect to the engine. This setting allows access to Tivoli Workload Scheduler from the Dynamic Workload Console. The authorization assigned to the user in the Tivoli Workload Scheduler security file determines the operations allowed.
If you want to test the connection to the Tivoli Workload Scheduler database (mandatory for managing reporting and event management functions), you must select Enable reporting and specify the user credentials.

5. Click Test Connection to check that the configuration was successful and that the Dynamic Workload Console is communicating with the selected engine. If the test connection fails, see Tivoli Workload Scheduler: Troubleshooting Guide, SC32-1275.

Note: Make sure you run “Post-installation steps to connect to Tivoli Workload Scheduler Version 8.3 Fix Pack 3” on page 225 before testing the engine connection if you are connecting to a Tivoli Workload Scheduler version 8.3 Fix Pack 3 engine or z/OS Connectors.

Quick steps to define a Dynamic Workload Broker connection:
The Dynamic Workload Console supports a single connection to one Dynamic Workload Broker engine at any given time for each authorized user. A different connection is supported for each authorized user.

After having logged in to the Dynamic Workload Console using the administrator user ID, or another user ID with assigned TDWBAAdministrator or TDWBConfigurator roles, follow these steps to create an engine connection to a supported Dynamic Workload Broker engine:

1. In the Dynamic Workload Console, click Dynamic Workload Broker to expand the tree.
2. Select Configuration.
3. Click Server connection.
4. In the Server Connection specify:

   Hostname
   The host name of the Dynamic Workload Broker you want to connect to.

   Non secure port
   The non-secure port to be used for connection.

   Secure port
   The secure port to be used for connection.

   Use Secure Connection
   Specify whether a secure connection must be used. For more information about security, see the Tivoli Workload Scheduler: Administration Guide, SC23-9113.

   Username
   Optionally specify a different user for the server connection. The connection to the new server is enabled using the credentials of the user you specified. Each user has access to only one server connection.

   Password
   Specify the password for the authenticated user the connection applies to.

5. Click OK to save your changes. The server connection you specified is enabled and is immediately effective.

Starting and stopping the Dynamic Workload Console:
To start and stop the Dynamic Workload Console or an engine, you must start and stop the application server instance it is installed on.
Embedded WebSphere Application Server on a Tivoli Integrated Portal in a Tivoli Workload Automation instance

If you installed the Dynamic Workload Console on the embedded WebSphere Application Server, you can start and stop the server as follows:

Windows operating system:
- To stop: `install_dir\wastools\stopWas.bat`
- To start: `install_dir\wastools\startWas.bat`

UNIX and Linux operating systems:
- To stop: `install_dir/wastools/stopWas.sh`
- To start: `install_dir/wastools/startWas.sh`

WebSphere Application Server on a Tivoli Integrated Portal outside a Tivoli Workload Automation instance

If you are using an external instance of WebSphere Application Server, use the following WebSphere Application Server scripts to start and stop an application server instance.

Note: These scripts can also be used to start and stop embedded WebSphere Application Server, although it is suggested that you use the method described above.

Windows operating system:
- `ewas_install_dir\bin\stopServer.bat app_server -user user_id -password user_id_pw`
- `ewas_install_dir\bin\startServer.bat app_server`

UNIX and Linux operating systems:
- `ewas_install_dir/bin/stopServer.sh app_server -user user_id -password user_id_pw`
- `/ewas_install_dir/bin/startServer.sh app_server`

where:
- `ewas_install_dir` is the directory where the WebSphere Application Server is installed.
- `app_server` is the server name specified in the Tivoli Integrated Portal profile related to the Dynamic Workload Console or to the engine. The default is `server1`.
- `user_id` is the administrator user ID specified when installing the Dynamic Workload Console or the engine.
- `user_id_pw` is the administrator user ID password specified when installing the Dynamic Workload Console or the engine.

Registry file information not found during upgrade:
You have tried to upgrade a stand-alone, fault-tolerant agent (an agent that is not shared with other components and does not have the connector feature) but the upgrade fails. If you were upgrading using the `twinst` script, you may have seen the following error message:

```
AWSFAB025E You are performing an update or uninstall operation, but the installation script has failed to find a previous instance of Tivoli Workload Scheduler in the Tivoli Workload Scheduler registry file. The script expected to find an entry belonging to the following user: `user_name`. and in the following registry file: `registry_file_name`.
```

If you were performing a silent installation, you may have seen the following error message:

```
AWSJIS165E No valid instance of Tivoli Workload Automation has been specified. Specify a valid instance or install the component in a new instance.
```

**Cause and solution**

This problem has occurred because of the following possible reasons:

- You have defined specified an incorrect installation path and the registry file cannot be found.
- You have used a user name that is not associated with the specific instance of Tivoli Workload Scheduler agent that you are upgrading.
- You are upgrading a stand-alone, fault-tolerant agent that has a corrupt registry file.

If you are sure you are using the correct installation path and user name, you can upgrade this agent without having to reinstall the product by using the Tivoli Workload Scheduler registry file recovery option, which re-creates the necessary files. See "Upgrading when there are corrupt registry files" on page 156 for the procedures on how to use the recovery option according to your upgrade method.

**The pobox files increase in size after you performed a parallel migration:**

After you migrate your environment, the pobox files increase in size.

**Cause and solution**

This problem occurs when performing a parallel migration for the following reasons:

- **In Step "Parallel 2: Switch the master domain manager to the new or upgraded backup master" on page 139** using the backup master domain manager V8.6 you define agent, pool, or dynamic pool workstations.
- **In Step "Parallel 5: Install a new master domain manager or upgrade your old master domain manager" on page 140** you did not set them to `ignore`, the agent, pool, or dynamic pool workstations you defined in Step "Parallel 2: Switch the master domain manager to the new or upgraded backup master" on page 139.

To solve the problem, perform the following steps:

1. From the backup master domain manager V8.6, set the workstation to `ignore`.
2. From the previous version master domain manager, run:
   ```
   JnextPlan -for 0000
   ```
Uninstallation problems
The following problems can occur when uninstalling:

- "An uninstallation on Windows fails"
- "An uninstallation fails during the restore profiles step, because the embedded WebSphere Application Server was not stopped"
- "The uninstallation of the Connector fails in the "Start the embedded WebSphere Application server" step"

An uninstallation on Windows fails:
You have tried to uninstall the product using the InstallShield wizard on a Windows workstation, but the uninstallation has failed.

Cause and solution
A possible cause of a failure of an InstallShield wizard uninstallation on Windows, is that the Services window of the Administrative Tools in the Control Panel is open.

Close the Services window. Rerun the uninstallation. If you have any problems, uninstall the services manually.

An uninstallation fails during the restore profiles step, because the embedded WebSphere Application Server was not stopped:
You have tried to uninstall the product but the uninstallation has failed at the restore profiles step. The error indicates that the embedded WebSphere Application Server has not stopped (if this fact is not reported by the log, check whether the application server has stopped, and if it has not, proceed in the same way).

Cause and solution
The problem is that the profiles cannot be restored while the application server is running, and the stop of the application server has failed. There are two possible causes:

- The Windows Service Control Manager was unable to stop the embedded WebSphere Application Server service before the timeout expired.
- The Windows Service Control Manager has given an error while trying to stop the service

Check the Windows system log files to see if an error is reported by the Windows Service Manager. If it has, you need to resolve the problem before continuing.

If no error is reported from the Windows Service Control Manager, it must be a timeout problem.

To solve the problem, do as follows:
1. Open the Windows Services panel
2. Stop the service that runs the embedded WebSphere Application Server from the panel
3. Close the Windows Services panel
4. Resume the installation from the restore profiles step.

The uninstallation of the Connector fails in the "Start the embedded WebSphere Application server" step:
You are uninstalling the Connector, but the uninstallation fails in the step: "Start the embedded WebSphere Application server", with the message:

AWSJIS038E An internal error has occurred. An unspecified internal error has occurred during the installation process.

---------- Error log ----------

Could not open service 'IBMWAS70Service - <TWS_user>'

Reason: The specified service does not exist as an installed service.

ADMU0116I: Tool information is being logged in file C:\Program Files\IBM\TWA\eWAS\profiles\TIPProfile\logs\server1\startServer.log

ADMU0128I: Starting tool with the TIPProfile profile

ADMU3100I: Reading configuration for server: server1

ADMU3028I: Conflict detected on port 28880. Likely causes: a) An instance of the server server1 is already running b) some other process is using port 28880

ADMU3027E: An instance of the server may already be running: server1

ADMU0111E: Program exiting with error: com.ibm.websphere.management.exception.AdminException: ADMU3027E: An instance of the server may already be running: server1

ADMU1211I: To obtain a full trace of the failure, use the -trace option.

ADMU0211I: Error details may be seen in the file: C:\Program Files\IBM\TWA\eWAS\profiles\TIPProfile\logs\server1\startServer.log

TWA_EXCEPTION

CMW3202E Command failed.

**Cause and solution**

This problem occurs when the WebSphere Application Server administration user ID and the TWS_user of the agent on which the Connector was installed are different. This can be caused by having installed the Dynamic Workload Console before the agent, and by not having installed the agent using the WebSphere Application Server administration user ID as the TWS_user ID.

To resolve the problem, do the following:

1. If you are using the interactive wizard, select the option to use the step restart facility. If the step failed in the silent installation, rerun the silent installation using the -resume option and omitting the -silent option so that the interactive wizard’s step restart facility can be used.

2. In the step restart facility, select the step that failed and put it into the Successful state.

3. Click Run next or Run all and let the wizard finish.
4. Follow the instructions in the *Tivoli Workload Scheduler: Administration Guide* for stopping and restarting the application server using the `stopWas` and `startWas` commands, ensuring to use the credentials of the WebSphere Application Server administration user.

**Fix pack installation problems**

This section describes problems and solutions for problems that might occur during the installation of a fix pack.

The following problem could be encountered:

**The update of the embedded WebSphere Application Server fails during the fix pack installation:**

You have tried to apply a fix pack, but the installation fails at the step "Update of the embedded WebSphere Application Server". A message similar to the following is given in the summary.log:

```
Updating bobcat
ERROR: The script ended abnormally. The reason is:
possible error, 65, launching updateinstaller.
The script exit code is 65
CMW3202E Command failed.
```

**Cause and solution**

More information can be found in the fix pack installation log. See "WebSphere Application Server installation log files" on page 33 for details.

One potential cause of the problem is disk space. Look for the following message in the fix pack installation log file:

```
(Aug 30, 2006 12:01:00 PM), UpdateInstaller,
com.ibm.ws.install.ni.ismp.actions.MaintenancePrereqCheckAction, err,
CWUPI0025E: There is insufficient free disk space on the system:
/TWS_home>/eWAS:
Required: 400 MB
Available: 146 MB
/tmp/:
Required: 250 MB
Available: 311 MB
```

Ensure there is enough free disk space on all required file systems and retry the operation.

**Security implications of the installation**

There are security implications involved in the installation of Tivoli Workload Scheduler, because some of the files used by the installation contain unencrypted passwords. The security exposure scenarios are as follows:

**Successful installation**

During a successful installation, some temporary files are written with unencrypted passwords. At the final step they are deleted. The exposure is the duration, or less, of the installation.
A failed installation which is resumed and finishes successfully.
   This is like the successful installation, except that the period of the
   exposure is lengthened by the time it takes you to fix whatever is the
   problem.

A failed installation which cannot be finished, and from which you have to
   recover manually
   In this case, the final step to delete the temporary files is not performed, so
   the files remain.

A silent installation
   You edit a response file, adding unencrypted passwords. The response file
   is not deleted, even after a successful installation. The exposure is
   permanent unless you delete the file.

The files where you can find unencrypted passwords are the following. They
   might not all be present, but you should check for all of them:

Windows operating system
   <TWS_home>\userdef_wnt
   <TEMP_DIR>\TWA/tws86\DB2Response.rsp
   <TEMP_DIR>\TWA/tws86\checkdb_root.sh
   <TEMP_DIR>\TWA/tws86\checkdbclient.sh

UNIX and Linux operating systems
   <TEMP_DIR>/TWA/tws86/DB2Response.rsp
   <TEMP_DIR>/TWA/tws86/checkdb_root.sh
   <TEMP_DIR>/TWA/tws86/checkdbclient.sh

In addition, on all platforms, the Tivoli Workload Scheduler response file if you
   used a silent installation (see the Tivoli Workload Scheduler: Planning and Installation
   Guide for details of the template response file names).

Verifying the installation
After installing the product, the installation process proceeds to complete, amongst
   other operations, the following configuration tasks:
   • Create the main local and global default settings.
   • Configure security access.
      A default operational Security file is created in the <TWS_home> directory. By
      default, it authorizes <TWS_user> and the administrator or root user. It is also
      updated when a full InstallShield wizard installation of the connector is
      performed.
   • Set workstation and user definitions each time you install or promote a Tivoli
      Workload Scheduler master domain manager.

To make sure that no errors occurred during installation, check the install log file
   (see “Installation log files” on page 32 for information about the log files and
   where to find them).

The following are examples of checks you can perform to verify the installation,
   and the corresponding recovery actions:

Check the main local and global default settings.
   If you promoted a workstation from the role of standard agent or
   fault-tolerant agents to the role of master domain manager, check that the
   master global option is set to the correct workstation name.
If it is wrong, you must manually edit the files to replace the current values with the correct ones.

**Check for the Security file**
Check that the default operational security file named Security was created in the `<TWA_home>\tws` directory. If this did not happen, create the file as follows:

1. Set the Tivoli Workload Scheduler environment by running the script `tws_env`.
2. Customize the Security file, as follows:
   a. Open the file `<TWS_home>/config/Security.conf`
   b. Edit the contents to reflect your environment and requirements
   c. Save the file as `<TWS_home>/Security.conf`
3. Run one of the following commands:
   - **Windows operating system**
     ```
     makesec Security.conf
     ```
   - **UNIX and Linux operating systems**
     ```
     makesec -l Security.conf
     ```

**Check for workstation and user definitions**
Check that your required workstation and user definitions are in place in the database of the master domain manager. To add missing definitions in the database, follow the instructions in the *Tivoli Workload Scheduler: User's Guide and Reference* or the online help for the Dynamic Workload Console.

**Uninstalling Tivoli Workload Scheduler manually**
This section describes how to manually remove an instance of Tivoli Workload Automation that did not completely uninstall.

The following are possible scenarios from which you might need to recover:
- You removed a previous installation of the product, but the uninstall procedure did not work properly and records of the previous installation were left on your system.
- For some reason the uninstallation as described in the *Tivoli Workload Scheduler: Planning and Installation Guide* does not work.
- Your installation fails and you cannot recover and finish the installation. In this event, you must determine which steps completed successfully, and start at the appropriate point in the uninstallation procedure. See "Correcting a failed step and continuing the installation" on page 186 for a detailed description of the steps.

The following provides details for uninstalling manually for the following operating systems:
- "Uninstalling manually on Windows operating system" on page 237
- "Uninstalling manually on UNIX" on page 239
- "Problems during manual uninstallation" on page 240

To remove an instance of Tivoli Workload Automation that contains an integrated installation of Tivoli Workload Scheduler and the Dynamic Workload Console, first perform the uninstallation of Tivoli Workload Scheduler as described in "Manually uninstall an integrated Dynamic Workload Console" on page 272. Then, to remove the Dynamic Workload Console, perform the following steps:
- On Windows operating system, perform steps 6 on page 272 and 7 on page 272.
- On UNIX and Linux operating systems, perform step 5 on page 273.

If you want to remove Tivoli Workload Scheduler from an instance of Tivoli Workload Automation without removing the Tivoli Workload Automation instance, contact IBM Software Support.

**Uninstalling manually on Windows operating system**

If Add or Remove Programs from the Windows Control Panel fails to uninstall Tivoli Workload Scheduler, perform the following steps:

1. If you have jobs that are currently running on the workstation, wait for them to finish. To determine which are not finished, check for jobs that are in the `exec` state. When there are no jobs in this state, and you have allowed sufficient time for all events to be distributed in your network, you can continue with the rest of the procedure.

2. Log on to the computer where Tivoli Workload Scheduler is installed as a user in the *Administrators* group.

3. From the `TWS_home/bin` directory run the following commands:
   ```
   conman "unlink workstation:noask"
   conman "stop;wait"
   conman "stopmon;wait"
   conman "shut;wait"
   ```

4. Stop the processes that are still active as follows:
   a. Open Services from the Windows Control Panel and stop the following Tivoli Workload Scheduler services:

   ```
   | Tivoli Netman for TWS_user |
   | Tivoli Token Service for TWS_user |
   | Tivoli Workload Scheduler for TWS_user |
   | Tivoli Workload Scheduler SSM Agent for TWS_user |
   | IBM Common Platform Agent for tws_cpa_agent_TWS_user |
   ```

   b. Run Windows Task Manager from the Windows Task Bar to end all the processes that are already running after stopping the Tivoli Workload Scheduler services.

   If the *End Process* action does not work, run the following steps from the `TWS_home/unsupported` directory:
   1) Run `listproc.exe`
   2) Read the PID number associated to the process that you want to end
   3) Run `killproc.exe <PID>`

5. Stop the WebSphere Application Server using the `conman stopappserver` command (see *Tivoli Workload Scheduler: User’s Guide and Reference*).

6. Use `wdrmvsp` to remove only the entries related to Tivoli Workload Scheduler. For information, see “Uninstalling using the Software Distribution CLI” on page 174. Open the `%WINDIR%\system32\TWSRegistry.dat` (for Windows 32 bit) or `%WINDIR%\TWSRegistry.dat` (for Windows 64 bit) file. Delete all the rows that contain the name of the `TWS_user`. For example, if the user ID is `twsuser`, delete the rows containing `twsuser`, as shown in the following example:
For a full description of the TWSRegistry.dat file, see “Registry file” on page 291.

7. Stop and remove the Tivoli Workload Scheduler services by issuing the following commands:

   TWA_home\eWAS\bin\WASService.exe -stop TWS_user
   TWA_home\eWAS\bin\WASService.exe -remove TWS_user

   Ensure that the Windows Services panel is closed when you do this.

8. Navigate to the install_dir and take note of the name of the .id file twainstancexxx.id. You need this information later in the procedure.

9. Delete the file teb_tws_cpa_agent_tws_user.ini, which is located in the %windir%\teb directory.

10. Delete the installation directory and all its contents.

   Note: If the manual deletion of these files is slow, see “File deletion on Windows too slow” on page 240.

11. Depending on what point the installation or uninstallation process reached before it failed, you might need to remove the Windows services. See the instructions for running regedit to do this in “Removing Windows registry keys” on page 243.

12. Remove the files:

   %WINDIR%\TWA\twainstancexxx.properties
   twainstancexxx.properties.ext

   where xxx is the name of the file you noted in step 8.

13. If you are performing this procedure because you are cleaning up a failed installation which could not be completed, you should also delete any files which contain unencrypted passwords.

    The files where you can find unencrypted passwords are the following. They might not all be present, but you should check for all of them:

    TEMP_DIR/tws86/checkdb_root.bat
    TEMP_DIR/tws86/checkdbclient.bat

    In addition, on all platforms, delete the Tivoli Workload Scheduler response file if you used a silent installation.

14. Depending on what point the installation or uninstallation process reached before it failed, you might need to remove the Add or Remove Programs keys. To do this, use the system’s facilities:
a. Open the Add or Remove Programs option window from the Windows Control Panel
b. If Tivoli Workload Scheduler is available on the menu, click Remove on it.
c. As you have, in the previous step, removed the uninstaller, a message is displayed, asking if you want to remove the Add or Remove Programs keys. Click "Yes" and the keys are removed.

15. Reboot the workstation to remove the services, any DLLs, any daemons, or any other executable programs from memory.

**Uninstalling manually on UNIX**

To uninstall manually, perform the following steps:

1. If you have jobs that are currently running on the workstation, wait for them to finish. To determine which are not finished, check for jobs that are in the exec state. When there are no jobs in this state, and you have allowed sufficient time for all events to be distributed in your network, you can continue with the rest of the procedure.

2. Log on to the computer where Tivoli Workload Scheduler is installed as root.

3. From the `<TWS_home>/bin` directory run the following commands:
   ```
   conman "unlink workstation;noask"
   conman "stop;wait"
   conman "stopmon;wait"
   conman "shut;wait"
   ```

4. From a shell script run the following command:
   ```
   ps -ef grep <TWS_install_dir>/bin/jobman
   ```
   This checks that the following processes are not active:

   | agent       |
   | JobManager  |
   | taskLauncher|
   | batchman    |
   | jobman      |
   | JOBMAN      |
   | mailman     |
   | monman      |
   | netman      |
   | ssmagent    |
   | stageman    |
   | writer      |

5. Stop the processes that are still active as follows:
   ```
   kill -9 <pid>
   ```

6. Stop the WebSphere Application Server using the `conman stopappserver` command (see *Tivoli Workload Scheduler: User’s Guide and Reference*).

7. Use `wdrmvsp` to remove only the entries related to Tivoli Workload Scheduler. For information, see "Uninstalling using the Software Distribution CLI" on page 174. Open `/etc/TWS/TWSRegistry.dat` and delete all the rows containing the TWS_user user ID. For example, if the user ID is twsuser, delete the rows containing twsuser, as shown in the following example:
For a full description of the TWSRegistry.dat file, see “Registry file” on page 291.

8. Navigate to the install_dir and take note of the name of the .id file twainstancexxx.id. You will need this information later in the procedure.

9. Delete the installation directory as follows:
   ```bash
   rm -R <TWS_home>
   ```

10. Delete the file /etc/teb/teb_tws_cpa_agent_tws_user.ini.

11. Delete the file:
    ```bash
    /etc/TWA/twainstancexxx.properties
    ```
    where xxx is the name of the file you noted in step 8.

12. If you are performing this procedure because you are cleaning up a failed installation which could not be completed, you should also delete any files which contain unencrypted passwords.
    The files where you can find unencrypted passwords are the following. They might not all be present, but you should check for all of them:
    ```bash
    TEMP_DIR/tws86/checkdb_root.sh
    TEMP_DIR/tws86/checkdbclient.sh
    ```
    In addition, on all platforms, delete the Tivoli Workload Scheduler response file if you used a silent installation.

13. Reboot the workstation to remove the services, any DLLs, any daemons, or any other executable programs from memory.

**Problems during manual uninstallation**

The following problem might occur during a manual uninstallation:

- “File deletion on Windows too slow”

**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.

**Uninstalling Tivoli Workload Scheduler connectors manually**

This section describes how to manually remove an instance of a Tivoli Workload Scheduler connector that did not completely uninstall.

The following are possible scenarios from which you might need to recover:
- You might have removed a previous installation of the connector, but the uninstall procedure did not work properly and records of the previous installation were left on your system.
- For some reason the uninstallation as described in **“Installing” on page 59** does not work.
- Your installation fails and you cannot recover and finish the installation. In this event, you must determine which steps completed successfully, and start at the appropriate point in the uninstallation procedure. See **“Correcting a failed step and continuing the installation” on page 186** for a detailed description of the steps.

The following provides details for uninstalling manually for the following operating systems:
• **“Uninstalling the connector manually on UNIX”**
• **“Uninstalling the connector manually on Windows operating system” on page 242**

**Uninstalling the connector manually on UNIX**

If you need to uninstall a connector manually, perform the following steps:

1. Log on to the computer where Tivoli Workload Scheduler is installed as **root**.
2. Access the directory: `<TWS_home>/wastools`
3. Stop the WebSphere Application Server using the **conman stopappserver** command (see **Tivoli Workload Scheduler: User’s Guide and Reference**)
4. Delete the installation directory as follows:
   ```sh
rmm -R `<TWS_home>`
   ```
   This step is not obligatory, but is just to save space on the file system. If you are in any doubt about risking deleting other Tivoli Workload Scheduler files, omit this step.
5. Open the `/etc/TWS/TWSZConnRegistry.dat` and delete all the rows containing the `<TWS_user>` user ID.
6. If you are performing this procedure because you are cleaning up a failed installation which could not be completed, you should also delete any files which contain unencrypted passwords.
   The files where you can find unencrypted passwords are the following. They might not all be present, but you should check for all of them:
   ```sh
   <TEMP_DIR>/tws86/checkdb_root.sh
   <TEMP_DIR>/tws86/checkdbclient.sh
   ```
   In addition, on all platforms, delete the Tivoli Workload Scheduler response file if you used a silent installation (see **Table 6 on page 84** for a details about response file names).
7. Remove the files:
   `<WINDIR>TWA	wainstancexxx.properties`
   `twainstancexxx.properties.ext`

**Uninstalling the connector manually on Windows operating system**

If you cannot run the add/remove program from the Windows Control Panel, perform the following steps:

1. Log on to the computer where Tivoli Workload Scheduler is installed as a user in the *Administrators* group.
2. Access the directory: `<TWS_home>/wastools`
3. Stop the WebSphere Application Server using the `conman stopappserver` command (see *Tivoli Workload Scheduler: User's Guide and Reference*)
4. Stop and remove the Tivoli Workload Scheduler services by issuing the following commands:
   ```
   <TWS_home>/eWAS\bin\WASService.exe -remove(<TWS_user>)
   <TWS_home>/eWAS\bin\WASService.exe -remove TWSZCONNECTOR(<TWS_user>)
   ```
   Ensure that the Windows Services panel is closed when you do this.
5. Open the `%WINDIR%\System32\TWSConnRegistry.dat` or `%WINDIR%\System32\TWSZConnRegistry.dat` file, depending on whether the connector is distributed or z/OS, and delete all the rows that contain the name of the `<TWS_user>`.
6. Delete the installation directory and all its contents. This step is not obligatory, but it saves space on the file system. If you are in doubt about deleting other Tivoli Workload Scheduler files, omit this step.
7. Depending on what point the installation or uninstallation process reached before it failed, you might need to remove the Add or Remove Programs keys. To do this, use the system's facilities:
   a. Open the Add or Remove Programs option window from the Windows Control Panel
   b. If the Tivoli Workload Scheduler Connector or the Tivoli Workload Scheduler for z/OS Connector is available on the menu, select it and click *Remove*.
   c. As you have, in the previous step, removed the uninstaller, a message is displayed, asking if you want to remove the Add or Remove Programs keys. Click “Yes” and the keys are removed.
8. If you are performing this procedure because you are cleaning up a failed installation which could not be completed, you should also delete any files which contain unencrypted passwords.
   The files where you can find unencrypted passwords are the following. They might not all be present, but you should check for all of them:
   ```
   <TWS_home>\userdef_wnt
   <TEMP_DIR>tws86/checkdb_root.sh
   <TEMP_DIR>tws86/checkdbclient.sh
   ```
   In addition, on all platforms, delete the Tivoli Workload Scheduler response file if you used a silent installation (see “Performing a silent installation” on page 84 for details of the template response file names).
9. Reboot the workstation to remove the services, any DLLs, any daemons, or any other executable programs from memory.
Removing Windows registry keys

During the life of this product it has undergone name changes and has been issued in a number of versions. There is more than one way to install and uninstall the product. All this leads to the risk that in upgrading from one version to another, one or more registry keys have been inadvertently left in the Windows Registry. This procedure is designed to help you remove these unwanted keys.

**Note:** If you make changes to the Windows Registry, you risk making the operating system unusable. You are strongly advised to back up the registry before you start.

A similar procedure is described in "Uninstalling manually on Windows operating system" on page 237; the same objective is achieved using different techniques.

This procedure is designed to identify and remove the keys for the following:

- Maestro versions 6.0 and 6.1
- Tivoli Workload Scheduler, versions 7.0, 8.1, 8.2.n, 8.3, and 8.4

Depending upon the version of Maestro or Tivoli Workload Scheduler, and the version of the operating system, some of the keys in this procedure might have already been removed by the InstallShield wizard uninstall program. If this is the case, skip that step and proceed to the next step in the procedure. Some of the names of the keys vary depending upon the choices made during the installation of Maestro or Tivoli Workload Scheduler, so make certain that you are aware of these original choices when locating the keys.

The procedure is as follows:

1. Stop Tivoli Workload Scheduler completely. The easiest way to do this is as follows:
   a. Change to the `<TWS_home>`\unsupported directory. In this directory are two files, listproc.exe and killproc.exe.
   b. Copy both of these files into the `<TWS_home>`\bin directory and set a path to `<TWS_home>` and `<TWS_home>`\bin, using the Windows path command.
   c. Type the following command:
      ```
      listproc | more
      ```
      A page that looks similar to this is displayed:

      | PID | Command    | # Handles | # Threads |
      |-----|------------|-----------|-----------|
      | 567 | agent      | 87        | 4         |
      | 687 | JobManager | 65        | 8         |
      | 443 | taskLauncher | 34      | 8         |
      | 624 | netman     | 86        | 5         |
      | 332 | tokensrv   | 62        | 8         |
      | 1088| writer     | 60        | 2         |
      | 5688| monman     | 89        | 3         |
      | 5364| ssmagent   | 210       | 19        |
      | 1052| mailman    | 85        | 2         |
      | 936 | batchup    | 57        | 4         |
      | 1020| batchman   | 92        | 2         |
      | 1036| JOBMAN     | 91        | 2         |
      | 1312| JOBMON     | 105       | 3         |

      This table shows the entire Tivoli Workload Scheduler process tree of a running fault-tolerant agents.
Note: There are other processes belonging to the operating system and other applications interspersed between these processes.

d. Write down the process ID (pid) of any TWS processes.

e. Stop the processes by issuing the following command for each running process: `killproc <pid>`. Killproc is a more reliable tool than the corresponding Microsoft tool, which does not always stop runaway processes.

2. Remove Tivoli Workload Scheduler using the InstallShield wizard uninstallation.

3. Reboot the workstation.

4. Select Start > Run, type `regedit` and press the enter key.

5. Select HKEY_LOCAL_MACHINE > Software.

6. Remove the keys from versions of Tivoli Workload Scheduler prior to version 8.3:
   a. Delete the Unison Software, Inc key.
   b. Close Software.
   c. Open System > CurrentControlSet > Services.
   d. Delete the maestro_<workstation>_<user_ID> key.
   e. Delete any key called netman_<system_name>_<user_ID>. DO NOT delete any other key called Netman. This Netman is part of Windows.
   f. Delete any key called tokensrv_<system_name>_<user_ID>. If you cannot locate a key of this type, look for keys called <process_name>_<user_ID> and delete them.

7. Remove the keys from Tivoli Workload Scheduler versions 8.3 and 8.4:
   a. Open System > CurrentControlSet > Services.
   b. Delete the tws_maestro_<user_ID> key.
   c. Delete the tws_netman_<user_ID> key. DO NOT delete any other key called Netman. This Netman is part of Windows.
   d. Delete the tws_tokensrv_<user_ID> key.
   e. Delete the tws_cpa_agent_<user_ID> key.
   f. Delete the tws_ssm_agent_<user_ID> key.

8. Close regedit.

9. Reboot the workstation.
Part 3. Dynamic Workload Console

Preparing

This chapter gives you an overview of what you need to know to prepare for
installation of the Dynamic Workload Console. It consists of the following sections:

- "Overview of the Dynamic Workload Console"
- "Installation overview"
- "Installation considerations" on page 246

Overview of the Dynamic Workload Console

The Dynamic Workload Console is a web-based user interface that is used with the
following set of products:

- Tivoli Workload Scheduler
- Tivoli Workload Scheduler for z/OS
- Tivoli Workload Scheduler for Applications
- Dynamic workload broker

You can access Tivoli Workload Scheduler and Dynamic Workload Broker
environments from any location in your network using one of the supported
browsers connected to the Dynamic Workload Console. The Dynamic Workload
Console must be installed on a system that can reach either the Tivoli Workload
Scheduler or the Dynamic Workload Broker nodes using network connections.

Installation overview

Perform the following steps to prepare, install, and configure the Dynamic
Workload Console:

1. Check the installation prerequisites at http://www.ibm.com/support/docview.wss?rs=672&uid=swg27020800 to verify that your system is compliant.
2. Collect the information necessary to fill in the required fields during the
   installation. See "Installing" on page 218.
3. Choose the installation method that best suits your needs as described in
   Installing.
4. Install the Dynamic Workload Console by following the instructions provided
   in "Installing the Dynamic Workload Console" on page 219.
5. If you plan to communicate with the Tivoli Workload Scheduler or Tivoli
   Workload Scheduler for z/OS Connector Version 8.3 Fix Pack 3, perform the
   post-installation steps as described in "Post-installation steps to connect to
   Tivoli Workload Scheduler Version 8.3 Fix Pack 3" on page 225.
6. Log in to the Dynamic Workload Console as described in "Accessing the
   Dynamic Workload Console" on page 226.
7. In the navigation tree on the left, click one of the following:
   - Tivoli Workload Scheduler
     To access the Tivoli Workload Scheduler available functions
   - Dynamic workload broker
     To access the Dynamic Workload Broker available functions
8. To effectively manage the functions available in the Dynamic Workload Console, create engine connections to the Tivoli Workload Scheduler and Dynamic Workload Broker environments that you want to manage. Without defining engine connections, you can use only a limited set of Dynamic Workload Console functions. For more information, see “Quick steps to define a Tivoli Workload Scheduler engine connection” on page 228 and “Quick steps to define a Dynamic Workload Broker connection” on page 229.

Installation considerations

Before you begin an installation or upgrade, consider the following items that might apply to your specific environment.

- Only one Dynamic Workload Console can be installed on a computer and can be installed as follows:
  - In a new Tivoli Workload Automation instance
  - In an existing Tivoli Workload Automation instance where the embedded WebSphere Application Server is already installed, but the Dynamic Workload Console is not installed
  - Outside any Tivoli Workload Automation instance, using an existing external instance of Tivoli Integrated Portal.

  For more information about Tivoli Workload Automation instances, see “Instances of Tivoli Workload Automation.”

- You cannot install more than one instance of the current version of the Dynamic Workload Console on the same workstation. If you attempt to install another instance of the Dynamic Workload Console onto a workstation that already has an upgradeable version on it, you will only be able to upgrade it.

- When you upgrade the Dynamic Workload Console, it is automatically upgraded into a new instance of Tivoli Workload Automation.

- If you plan to install the Dynamic Workload Console on already-installed Tivoli Integrated Portal, ensure that the server associated to the profile where you plan to install is active before starting the installation. Only profiles that are created as described and without customization are supported.

- You must restart the Dynamic Workload Console immediately after the installation if you plan to connect to Internet Protocol version 6 (IPv6) enabled engines.

- Before installing Dynamic Workload Console on Windows and Windows 64, you must start the workstation service of Windows. This applies to Windows 2003 and Windows 2008.

Selecting your installation method

You can install the Dynamic Workload Console using one of the following methods:

Launchpad

Use the launchpad to guide you through the installation of the Dynamic Workload Console, and the Tivoli Workload Scheduler components, from a single interface. For more information about how to install using the launchpad, see “Installing from the launchpad.”

Installation wizard

Access the installation wizard by running the appropriate setup command and entering the configuration settings to install and configure your installation. Using this method, you can synchronously monitor the installation processing and results. For more information, see “Using the installation wizard” on page 219.
This method of installation uses a Java Virtual Machine, and therefore has specific system requirements. See the Dynamic Workload Console System Requirements Document at http://www.ibm.com/support/docview.wss?rs=672&uid=swg27020800 for details on installation requirements.

Silent mode

Customize a response file by adding all configuration settings to be used during installation, and then invoke from the command line the setup command using the -silent keyword. Using this method, you can run the installation unattended and in the background. For more information, see “Performing a silent installation” on page 223.

Instances of Tivoli Workload Automation

During the installation of the Dynamic Workload Console you must decide whether to install into an existing instance of Tivoli Workload Automation or whether to create a new instance. For information, see Instances of Tivoli Workload Automation.

If you are installing into an existing instance of Tivoli Workload Automation, you can install certain components, depending on the components or products that currently exist in that instance. Table 18 describes the actions that you can perform in each different scenario.

Table 18. Installing into an existing instance of Tivoli Workload Automation

<table>
<thead>
<tr>
<th>If the existing Tivoli Workload Automation instance contains:</th>
<th>You can perform the following:</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Dynamic Workload Console version 8.4, 8.5, or 8.5.1</td>
<td>Upgrade</td>
</tr>
<tr>
<td>A Dynamic Workload Console version 8.4, 8.5, or 8.5.1 installed on external WebSphere Application Server</td>
<td>Uninstall and reinstall the Dynamic Workload Console. It is not possible to upgrade the Dynamic Workload Console in this case.</td>
</tr>
<tr>
<td>A Dynamic Workload Console version 8.6</td>
<td>Take no action. It is not possible to install the Dynamic Workload Console in this case.</td>
</tr>
<tr>
<td>Tivoli Workload Scheduler version 8.5 or 8.5.1 master domain manager or backup domain manager</td>
<td>Take no action. It is not possible to install the Dynamic Workload Console in this case.</td>
</tr>
<tr>
<td>A Tivoli Workload Scheduler version 8.6 master domain manager or backup domain manager</td>
<td>Install the Dynamic Workload Console on the common embedded WebSphere Application Server.</td>
</tr>
<tr>
<td>A Tivoli Workload Scheduler version 8.5 or 8.5.1 agent with connector</td>
<td>Take no action. It is not possible to install a second instance of the Dynamic Workload Console on the same computer.</td>
</tr>
<tr>
<td>A Tivoli Workload Scheduler version 8.6 agent with connector</td>
<td>Install the Dynamic Workload Console on the common embedded WebSphere Application Server.</td>
</tr>
<tr>
<td>A Tivoli Workload Scheduler for z/OS connector version 8.5 or 8.5.1</td>
<td>Take no action. It is not possible to install a second instance of the Dynamic Workload Console on the same computer.</td>
</tr>
<tr>
<td>A Tivoli Workload Scheduler for z/OS connector version 8.6</td>
<td>Install the Dynamic Workload Console on the common embedded WebSphere Application Server.</td>
</tr>
</tbody>
</table>
Table 18. Installing into an existing instance of Tivoli Workload Automation (continued)

<table>
<thead>
<tr>
<th>If the existing Tivoli Workload Automation instance contains:</th>
<th>You can perform the following:</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Tivoli Workload Scheduler version 8.6 dynamic domain manager or backup dynamic domain manager.</td>
<td>Install the Dynamic Workload Console on the common embedded WebSphere Application Server.</td>
</tr>
</tbody>
</table>

Any components that are not mentioned in this table, such as the agent without connector, the domain manager or the command-line client, are not installed in Tivoli Workload Automation instances, and so do not impact the Dynamic Workload Console installation.

**Installation media**

The Dynamic Workload Console is packaged into multiple DVDs, one for each of the supported operating systems. Each DVD contains:
- The installable image
- The `setup` file
- The sample response files
- The launchpad

For a complete list of DVDs and supported operating systems, see the Dynamic Workload Console downloadable documentation at [http://www.ibm.com/support/docview.wss?rs=672&uid=swg24029125](http://www.ibm.com/support/docview.wss?rs=672&uid=swg24029125).

**Note:**

1. If you copy or mount the DVD to a system directory, make sure that the path name to that directory does not contain the following unsupported characters: { } [ ] < > $ ! ? ! * + " / % ’ or non US-ASCII characters.
2. If you plan to install on a Windows system from a mapped remote drive, make sure you map the remote folder locally on the system where you want to install, and then run the installation using the local path.
3. If you plan to install on Linux, make sure that the files contained in the mounted image have executable permission, and that the `SETUP.bin` file is not located in a path with blanks.

**Installation log files**

The type of log files you find on your system depends on the type of installation you performed. This section describes the logs associated with the different installations.

For more information about log files, see the Administration Guide.

**Interactive wizard installation and uninstallation log files:**
You can check the following log files for information about the installation. Details of the installation process are recorded in log files on the local computer in the following directories:

**Note:** The following values are valid only if you have not changed the default value of the TEMP system variable.

**Windows operating system**

```
%Temp%\TWA\tdwc86
```

**UNIX and Linux operating systems**

```
/tmp/TWA/tdwc86
```
Table 19 lists the InstallShield wizard log files.

Table 19. Installation log files

<table>
<thead>
<tr>
<th>Log file name</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>tdwcstatus.log</td>
<td>Dynamic Workload Console installation status log file. It reports if the installation completed successfully or with errors. In case of errors it indicates if the error is due to an incorrect field value or to a failed step.</td>
</tr>
<tr>
<td>tdwcininstall.log</td>
<td>Dynamic Workload Console installation log file</td>
</tr>
<tr>
<td>tdwcnuninstall.log</td>
<td>Dynamic Workload Console uninstallation log file</td>
</tr>
<tr>
<td>securityConfignnn.log</td>
<td>Dynamic Workload Console log file containing details about the Tivoli Integrated Portal security configuration performed during installation. The numeric value nnnn is automatically assigned. Access the tdwcininstall.log file to see the filename of the securityConfignnn.log file.</td>
</tr>
<tr>
<td>wsadmin.log</td>
<td>Dynamic Workload Console log file containing details about the interaction of the installation with WebSphere Application Server.</td>
</tr>
<tr>
<td>TIPInstaller-00.log</td>
<td>Tivoli Integrated Portal installation log file.</td>
</tr>
</tbody>
</table>

For multiple installations on the same workstation, the log header and footer indicate the user ID (TWS_user) for which the installation was performed.

**Note:** If you are running a silent installation and the response file you are using does not have the correct syntax, the installation fails without producing a log.

**Installation log files for the embedded WebSphere Application Server:**

The application server installation has no log. However, if you update the application server, for example during the application of a Tivoli Workload Scheduler fix pack, a log is created which gives information about the update. The log can be found in the directory `TWS_home/eWAS/logs/update`, where you will find a directory that identifies the fix pack that has been installed, for example: `7.0.0-WS-WASEmbeded-AixPPC64-FP0000027.install`, which contains a log file called `updatelog.txt`.

The log for the startup of the application server can be found at:

`TWS_home/eWAS/profiles/TIPProfile/logs/server1/startServer.log`

**Installing**

This chapter describes how to install the Dynamic Workload Console. It is divided into the following sections:

- “Installing the Dynamic Workload Console” on page 219
- “Post-installation steps to connect to Tivoli Workload Scheduler Version 8.3 Fix Pack 3” on page 225
- “Post-installation steps to configure the use of Lightweight Third-Party Authentication (LDAP)” on page 226
- “Accessing the Dynamic Workload Console” on page 226
- “Starting and stopping the Dynamic Workload Console” on page 229
Installing the Dynamic Workload Console

This section explains how to install the Dynamic Workload Console using the available installation methods. It is divided into the following topics:

- "Using the launchpad" on page 219
- "Using the installation wizard" on page 219
- "Performing a silent installation" on page 223
- "Installing the Tivoli Integrated Portal on an external WebSphere Application Server from the images" on page 224

Using the launchpad

You can install the Dynamic Workload Console using the launchpad. Use the instructions for launching and running the launchpad at Installing from the launchpad and choose the Dynamic Workload Console installation option in the launchpad. Follow the on-screen instructions. The launchpad runs the installation wizard with some of the options pre-filled. Follow the instructions for "Using the installation wizard" on page 219, to complete the process.

Using the installation wizard

Follow these steps to install the Dynamic Workload Console using the installation wizard:

1. Browse to the setup directory and start the installation by running the setup file. The installation wizard first checks if there is enough free space available in the Java temporary directory. If not, the installation exits, and you must increase the size of the Java temporary directory, as described in the Tivoli Workload Scheduler System Requirements Document at http://www.ibm.com/support/docview.wss?rs=672&uid=swg27019747, before rerunning the installation wizard.

2. Select the language to use while installing the Dynamic Workload Console, and click OK.

3. In the welcome panel, click Next to continue with the installation.

4. Read and accept the license agreement. Click Next.

5. Select the Tivoli Integrated Portal instance. Choose among the following:
   - If you choose to install a new instance of the Tivoli Integrated Portal or if you choose to install on an existing instance of Tivoli Workload Automation that does not contain the embedded WebSphere Application Server, perform the steps in "Installing a new instance of the Tivoli Integrated Portal" on page 220, Perform this installation if you do not have the Tivoli Integrated Portal already installed or if you have installed a Tivoli Workload Scheduler component that does not install the embedded WebSphere Application Server for example a fault-tolerant agent.
   - If you choose to install on an existing Tivoli Workload Automation instance that contains the embedded WebSphere Application Server, perform the steps in "Installing on an existing instance of the embedded WebSphere Application Server" on page 222, Perform this installation if you have already installed a Tivoli Workload Automation component that installs the embedded WebSphere Application Server also, for example the master domain manager.
   - If you choose to install on top of your existing instance of Tivoli Integrated Portal, perform the steps in "Installing on your existing instance of Tivoli Integrated Portal" on page 223, You perform this installation if you have already installed a Tivoli Integrated Portal with another Tivoli product. For a list of supported Tivoli Integrated Portals, see the Tivoli Workload Scheduler
Starting from V8.6 we do not support anymore the Dynamic Workload Console installed on external WebSphere Application Server. If you do not have the Tivoli Integrated Portal installed, you can install it using the installation DVD or the appropriate images as described in the "Installing the Tivoli Integrated Portal on an external WebSphere Application Server from the images" on page 224.

6. Select an installation location. Click **Next**.

7. Specify the user name and password of the Tivoli Integrated Portal user that you want to use as the Dynamic Workload Console administrator.

   **Note:** The user name and password must be operating systems credentials. On Windows operating systems, if the user name you specify does not exist, a new operating system user is created.

   The user name must be unique, 3 to 60 characters in length, and contain only the characters a-z, A-Z, 0-9, period (.), hyphen (-), underscore (_), and double-byte character set (DBCS) characters.

   The password must be 5 to 16 characters in length and contain only the characters a-z, A-Z, 0-9, period (.), hyphen (-), and underscore (_).

   The user must be created as described in the **Rules for using a Federated User Registry with Tivoli Workload Scheduler**.

   Confirm the password and click **Next**.

8. Choose a new path to install into or choose the path of the existing Tivoli Workload Automation instance. Choose the path where you want to install, from now on referred to as `twa_install_dir`, or accept the default path, and click **Next**.

   Make sure that the installation path is 32 characters or less in length and that it does not contain special characters.

**Installing a new instance of the Tivoli Integrated Portal:**

The following applies if you are installing a new Tivoli Workload Automation instance or if you are installing over an existing Tivoli Workload Automation instance where the embedded WebSphere Application Server has not yet been installed. You are in this case if you do not have the Tivoli Integrated Portal already installed or if you have installed a Tivoli Workload Scheduler component that does not install the embedded WebSphere Application Server for example a fault-tolerant agent. In this case Tivoli Workload Scheduler installs the embedded WebSphere Application Server and the Tivoli Integrated Portal.

Follow these steps if you selected to install the Tivoli Integrated Portal and the Dynamic Workload Console:

In the installation choice window, select one of the following installation types.

**Default Installation**

If you want to use the default Tivoli Integrated Portal settings, proceed with the installation as described in "Default installation" on page 221.

**Advanced Installation**

If you want to customize the Tivoli Integrated Portal settings, proceed with the installation as described in "Advanced installation" on page 221.

**Default installation:**

Follow these steps to proceed with a default installation:
1. To start the installation, check that the values displayed in the installation summary window are correct and click **Install**.

2. When the installation completes successfully, a window opens showing links to the user interface on the Tivoli Integrated Portal. For more information, see "Accessing the Dynamic Workload Console” on page 226. If the installation fails, the window contains the list of the items that were not installed and the location of the log file. Click **Finish**.

**Advanced installation:**
Perform the following steps to proceed with an advanced installation:

1. Specify the following port numbers for the Tivoli Integrated Portal or accept the default values. These are embedded WebSphere Application Server ports used by Tivoli Integrated Portal.

   **HTTP transport**
   The number of the port that the portal uses for HTTP transport. The default value is **29080**.

   **HTTPS transport**
   The number of the port that the portal uses for secure HTTP transport (HTTPS). The default value is **29443**.

   **Bootstrap**
   The port number for the bootstrap function. The default value is **22809**.

   **SOAP connector**
   The port number for the Simple Object Access Protocol (SOAP) connector on the portal. The default value is **28880**.

   **SAS server authentication listener**
   The SAS SSL server authentication listener port number on the portal. The default value is **29401**.

   **CSIv2 server authentication listener**
   The CSIv2 SSL ServerAuth Listener port number on the portal. The default value is **29403**.

   **CSIv2 Client Authentication Listener**
   The CSIv2 SSL ClientAuth Listener port number on the portal. The default value is **29402**.

   **ORB listener**
   The ORB listener port number on the portal. The default value is **29100**.

   **Administrative console**
   The HTTP administrative console port on the portal. The default value is **29060**.

   **Administrative console secure**
   The HTTP administrative console secure port on the portal. The default value is **29043**.

   **IPC connector**
   The IPC connector on the portal. The default value is **29314**.

   **REST notification**
   The REST notification port on the portal. The default value is **29324**.

   **DCS Unicast port**
   The DCS Unicast port on the portal. The default value is **29353**.

   Click **Next**.
2. Complete the installation by following the steps described in "Default installation" on page 221.

Installing on an existing instance of the embedded WebSphere Application Server:
You perform this installation if you have already installed a Tivoli Workload Automation component that installs the embedded WebSphere Application Server also, for example the master domain manager. Follow these steps to install the Dynamic Workload Console on an existing instance of the embedded WebSphere Application Server:
1. Select the existing Tivoli Workload Automation directory.
2. Supply the username and password of the existing instance of the embedded WebSphere Application Server.

   **Note:** If you have already installed WebSphere Application Server into your existing Tivoli Workload Automation instance but do not know the username, click Retrieve. The username is retrieved but you still must provide the password. This operation may take a few minutes. If you are performing a silent installation, to find these credentials, run showSecurityProperties before running the installation.

3. Select if you want the administrator to access the Tivoli Workload Scheduler console, the Dynamic Workload Broker console, or both. Click Next.

   **Note:** If you select one of the two available user interfaces, after installing you can authorize the user to access the other user interface by assigning him one of the predefined roles created by the installation process. For more information, see the information about configuring the Dynamic Workload Console in the Tivoli Workload Scheduler: Administration Guide.

4. To start the installation, check that the values displayed in the installation summary window are correct and click Install.

   Specify the following port numbers for the Tivoli Integrated Portal or accept the default values. These are embedded WebSphere Application Server ports used by Tivoli Integrated Portal

   **IPC connector**
   The IPC connector on the portal. The default value is 29314.

   **REST notification**
   The REST notification port on the portal. The default value is 29324.

   **DCS Unicast port**
   The DCS Unicast port on the portal. The default value is 29353.

5. When the installation completes successfully, a window opens showing links to the user interface on the Tivoli Integrated Portal. For more information, see "Accessing the Dynamic Workload Console" on page 226. If the installation fails, the window contains a list of the items that were not installed and the location of the log file. Click Finish.

Installing on your existing instance of Tivoli Integrated Portal:
You perform this installation if you have already installed a Tivoli Integrated Portal with another Tivoli product. Follow these steps to install the Dynamic Workload Console on top of an existing Tivoli Integrated Portal instance:
1. Select the existing Tivoli Integrated Portal instance over which you want to install the Dynamic Workload Console by specifying the installation path.
2. Specify the user ID and password of an existing Tivoli Integrated Portal user that you want to set as the Dynamic Workload Console administrator.
Note: If you select one of the two available user interfaces, after installing you can authorize the user to access the other user interface by assigning him one of the predefined roles created by the installation process. For more information, see the information about configuring the Dynamic Workload Console in the Tivoli Workload Scheduler: Administration Guide.

3. To start the installation, check that the values displayed in the installation summary window are correct and click Install.

4. When the installation completes successfully, a window opens showing links to the user interface on the Tivoli Integrated Portal. For more information, see "Accessing the Dynamic Workload Console" on page 226. If the installation fails, the window contains a list of the items that were not installed and the location of the log file. Click Finish.

Performing a silent installation

You can run the installation in unattended mode from the command line by adding the -silent parameter when running the setup installation file. Perform the following steps:

- Run the installation as root on UNIX operating systems, or as Administrator on Windows operating systems.
- Specify all the settings that are prompted when installing using the installation wizard.

The installation settings are provided using a response file.

Edit the response file templates provided on the installation DVDs in the \tdwc\responsefiles\ directory. Instructions for customizing the files are included in the files as commented text. For details about response file properties, see "The Dynamic Workload Console response file properties" on page 303.

Table 17 on page 223 lists the response files and the types of installation each performs by operating system:

<table>
<thead>
<tr>
<th>Type of installation</th>
<th>Response file to use on Unix</th>
<th>Response file to use on Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fresh Dynamic Workload Console on existing TWA instance</td>
<td>TDWC86_FRESH_existTWA_UNIX.txt</td>
<td>TDWC86_FRESH_existTWA_WIN.txt</td>
</tr>
<tr>
<td>Fresh Dynamic Workload Console on an external Tivoli Integrated Portal</td>
<td>TDWC86_FRESH_extTIP_UNIX.txt</td>
<td>TDWC86_FRESH_extTIP_WIN.txt</td>
</tr>
<tr>
<td>Fresh Dynamic Workload Console on a new TWA instance</td>
<td>TDWC86_FRESH_newTWA_UNIX.txt</td>
<td>TDWC86_FRESH_newTWA_WIN.txt</td>
</tr>
<tr>
<td>Uninstall the Dynamic Workload Console</td>
<td>TDWC86_UNINSTALL.txt</td>
<td>TDWC86_UNINSTALL.txt</td>
</tr>
<tr>
<td>Upgrade the Dynamic Workload Console on existing Tivoli Workload Automation instance (embedded WebSphere Application Server)</td>
<td>TDWC86_UPGRADE_embeddedWAS_UNIX.txt</td>
<td>TDWC86_UPGRADE_embeddedWAS_WIN.txt</td>
</tr>
</tbody>
</table>
Note: In the upgrade scenarios, choose the embedded version of IBM Websphere Application Server that you originally chose when you installed the Dynamic Workload Console version 8.4 or higher.

To install in silent mode, perform these steps on the computer on which you want to install the Dynamic Workload Console:

1. Copy the sample response file for that operating system to a local temporary directory.
2. Customize the options contained in the response file to suit your requirements and environment. For information about the available options, see "The Dynamic Workload Console response file properties" on page 303.
3. Run the following command:
   - Windows operating system:
     ```
     SETUP.exe -options response_file.txt -silent
     ```
   - UNIX and Linux operating systems:
     ```
     ./SETUP.bin -options response_file.txt -silent
     ```
   where response_file is the full path name.
4. Check the result of the silent installation as follows:
   - Windows operating system:
     The installation command is asynchronous, meaning that when it is issued it starts an installation procedure and then ends without returning any value or message. To know whether or not the silent installation ran successfully, see the installation result reported in the tdwcinstall.log installation log file stored in the temporary directory.
   - UNIX and Linux operating systems:
     The installation command is synchronous and it returns 0 if the installation ran successfully, or a nonzero value if the installation failed.

Note: For information about the installation result, see the tdwcinstall.log installation log file stored in the temporary directory.

Installing the Tivoli Integrated Portal on an external WebSphere Application Server from the images

The following procedure applies if you do not have the Tivoli Integrated Portal installed but you have the WebSphere Application Server installed. To install the Tivoli Integrated Portal, perform the following steps:

1. From the installation DVD or from the downloaded images, go to the TDWC_<operating_system>\TDWC\WEBUI\<operating_system>\TIP\ directory where the sample_response.txt file is located.
2. Follow the instructions provided in the sample_response.txt file to customize the properties necessary to perform the installation.
3. Run the following command:
   ```
   install.sh/bat <java_jre_16_home> sample_response.txt
   ```
   where java_jre_16_home is the path where the Java V16 is installed.

The Tivoli Integrated Portal installation creates the TIPProfile profile into your existing instance of the WebSphere Application Server. After you installed the Tivoli Integrated Portal, you can install the Dynamic Workload Console on this new Tivoli Integrated Portal instance by following the instructions provided in "Installing the Dynamic Workload Console" on page 219.
Post-installation steps to connect to Tivoli Workload Scheduler Version 8.3 Fix Pack 3

To access a Tivoli Workload Scheduler Version 8.3 Fix Pack 3 environment, you must enable Tivoli Workload Scheduler to work with the Dynamic Workload Console.

Note:

1. These steps are not necessary to connect to a Tivoli Workload Scheduler environment for any version higher than V8.3 Fix Pack 3. Any upgrades performed after version 8.3 Fix Pack 3 will maintain any changes made during this procedure.

2. If you plan to communicate from the Dynamic Workload Console version 8.4 or higher to Tivoli Workload Scheduler, Version 8.3 Fix Pack 3, make sure that the APAR PK47309 is installed on top of the Tivoli Workload Scheduler engine. For more information, contact IBM Software Support.

3. Before proceeding, it is recommended that you run the backupConfig.sh or backupConfig.cmd script to backup the Tivoli Workload Scheduler configuration. For information about how to run these scripts, see the Tivoli Workload Scheduler: Administration Guide.

This task must be run on the system where the Tivoli Workload Scheduler engine that you want to connect to is installed:

**Tivoli Workload Scheduler distributed environment**
- On the master domain manager.
- On a full status fault-tolerant agents (FTA) workstation where the Tivoli Workload Scheduler connector is installed.

**Tivoli Workload Scheduler z/OS environment**
- On the distributed system where the Tivoli Workload Scheduler z/OS Connector is installed.

Perform the following steps:

1. Make sure that the embedded or external version of WebSphere Application Server, as appropriate, is started on the Tivoli Workload Scheduler workstation and then run the following script:

   **On Windows operating system:**
   As Administrator, from the directory `TWS_home\wastools`:
   ```bash
   webui -operation enable -user TWS_user -password TWS_user_pw
   -port TWS_port [-server TWS_server]
   ```

   **On UNIX and Linux systems**
   As root, from the directory `TWS_home/wastools`:
   ```bash
   ./webui.sh -operation enable -user TWS_user -password TWS_user_pw
   -port TWS_port [-server TWS_server]
   ```

   where:
   - `TWS_user`
     The Tivoli Workload Scheduler administrator user ID.
   - `TWS_user_pw`
     The Tivoli Workload Scheduler administrator password.
TWS_port
The SOAP port of the WebSphere Application Server where the Tivoli Workload Scheduler is installed. This is a mandatory setting when using the enable flag. Its default values are 31118 for distributed environments, and 31128 for z/OS environments.

TWS_server
The name of the server specified in the WebSphere Application Server profile used by Tivoli Workload Scheduler. By default the value assigned to this field is server1.

2. Stop and start the external or embedded WebSphere Application Server on the Tivoli Workload Scheduler system where you run the script.

When you have completed these steps, you are ready to create engine connections for the Tivoli Workload Scheduler workstation and to manage your Tivoli Workload Scheduler production environment. For information about how to accomplish these tasks, access the Dynamic Workload Console online general help.

Post-installation steps to configure the use of Lightweight Third-Party Authentication (LDAP)
If the Dynamic Workload Console and the Tivoli Workload Scheduler engine or the Tivoli Workload Scheduler z/OS Connector have been configured with the same LDAP user registry, or are installed on the same computer, you might receive a connection failure. If this happens, use the same Lightweight Third-Party Authentication (LTPA) keys on all servers: the Dynamic Workload Console, the Tivoli Workload Scheduler engine server, and the Tivoli Workload Scheduler z/OS Connector server.

To align the LTPA keys, see the section on configuring the use of Lightweight Third-Party Authentication in the Administration Guide.

Accessing the Dynamic Workload Console
When the installation of the Dynamic Workload Console completes successfully, a message with links to the Integrated Solutions Console portal is displayed. If you used the silent installation, this information is stored in the tdwinstall.log installation log file. For more details about where to find the installation logs, see "Interactive wizard installation and uninstallation log files" on page 248.

From a supported browser, access one of the following links provided by the installation program:

http://dynamic_workload_console_system:http_port/ibm/console
https://dynamic_workload_console_system:https_port/ibm/console

where:
dynamic_workload_console_system
The hostname or IP address of the system where you installed the Dynamic Workload Console.

http_port
The port number used to access the Dynamic Workload Console using an unsecure connection over HTTP. The default value for this port number is 29080 if you installed the Dynamic Workload Console in a new Tivoli Workload Automation instance. If you installed the Dynamic Workload
Console into an existing Tivoli Workload Automation instance, the value for this port is inherited. If the existing Tivoli Workload Automation instance contains the current version of Tivoli Workload Scheduler using default ports, the value is 31123.

https_port
The port number used to access the Dynamic Workload Console using a secure connection over HTTPS. The default value for this port number is 29443 if you installed the Dynamic Workload Console as a new Tivoli Workload Automation instance. If you installed the Dynamic Workload Console into an existing Tivoli Workload Automation instance, the value for this port is inherited. If the existing Tivoli Workload Automation instance contains the current version of Tivoli Workload Scheduler using default ports, the value is 31124.

When connecting to the Tivoli Integrated Portal using an HTTPS connection, if you receive a security alert, proceed with the Dynamic Workload Console working session. If you receive security information windows while navigating through the Tivoli Integrated Portal, choose to display nonsecure items to proceed. If you are using Internet Explorer, you can prevent these windows from opening by setting Display mixed content to Enable in the Security settings.

In the Tivoli Integrated Portal login portlet, enter the user ID and password you specified during the installation, and click Log in.

On the navigation bar on the left, expand the Tivoli Workload Scheduler entry to access the Dynamic Workload Console and then the Tivoli Workload Scheduler components. Expand the Dynamic Workload Broker entry to access Dynamic Workload Broker environments.

To effectively use the functions of these two products, you must define connections to the Tivoli Workload Scheduler engines and the Dynamic Workload Broker servers.

Without defining engine connections, you can perform only this limited set of operations:

On Tivoli Workload Scheduler
- Create browse tasks
- Create report tasks
- Create event management tasks
- Define user preferences

On Dynamic Workload Broker
Define user preferences

If the user ID you used to connect to the Dynamic Workload Console has been assigned a role different from TWSWEBUIAdministrator and TDWBAAdministrator, you will see a subset of the available panels. This subset depends on the authorizations assigned to the role associated to your user ID. For more information about roles, see the information about configuring the Dynamic Workload Console in the Tivoli Workload Scheduler: Administration Guide.

If the user ID you used to connect to the Dynamic Workload Console has no role assigned, you do not see the entries for Tivoli Workload Scheduler and Dynamic Workload Broker in the Tivoli Integrated Portal portal navigation tree.
Quick steps to define a Tivoli Workload Scheduler engine connection

After logging in to the Dynamic Workload Console using the administrator userid or another userid with assigned TWSWEBUIAdministrator or TWSWEBUIConfigurator roles, use the following steps to create an engine connection to one of your supported Tivoli Workload Scheduler engines.

**Note:** If you installed the Dynamic Workload Console into a Tivoli Workload Automation instance that had the embedded WebSphere Application Server already installed, the connection to the Tivoli Workload Scheduler component (for example, master domain manager, backup master domain manager, or connector) is automatically defined with blank credentials. The connection is shared with all the Dynamic Workload Console users and no further credentials are needed because Single Sign On is automatically implemented for the component. The same situation applies if you install a Tivoli Workload Scheduler component into a Tivoli Workload Automation instance where the Dynamic Workload Console and the embedded WebSphere Application Server are already installed.

1. To expand the tree, click the Dynamic Workload Console and Tivoli Workload Scheduler.
2. Select Quick start
3. Click New Engine.
4. In the Engine Connection Properties window, assign a name to the engine connection and specify:
   
   **Engine Type**
   
   Either z/OS or Distributed. This is the type of the Tivoli Workload Scheduler engine to connect to.

   **Hostname**
   
   The hostname or IP address of system where the distributed engine or the z/OS connector, for z/OS engine types, runs.

   **PortNumber**
   
   The bootstrap port number for the Tivoli Workload Scheduler engine. Default values are 31117 for distributed engine, and 31217 for z/OS connector.

   **Remote Server Name**
   
   This setting is valid and mandatory only for z/OS engines. It is the value specified when the engine was created in the z/OS connector. It must exactly match the z/OS connector engine name and is case sensitive. If the engine was defined using the WASTOOLS "createZosEngine" COMMAND, this is the value specified in the -name parameter. This is the name of the remote server of the engine as it is specified in the z/OS connector.

   **Userid and Password**
   
   The user ID and password that are used to connect to the engine. This setting allows access to Tivoli Workload Scheduler from the Dynamic Workload Console. The authorization assigned to the user in the Tivoli Workload Scheduler security file determines the operations allowed.

If you want to test the connection to the Tivoli Workload Scheduler database (mandatory for managing reporting and event management functions), you must select Enable reporting and specify the user credentials.
5. Click **Test Connection** to check that the configuration was successful and that
the Dynamic Workload Console is communicating with the selected engine. If
the test connection fails, see *Tivoli Workload Scheduler: Troubleshooting Guide,
SC32-1275*.

**Note:** Make sure you run "Post-installation steps to connect to Tivoli Workload
Scheduler Version 8.3 Fix Pack 3" on page 223 before testing the engine
connection if you are connecting to a Tivoli Workload Scheduler version
8.3 Fix Pack 3 engine or z/OS Connectors.

**Quick steps to define a Dynamic Workload Broker connection**

The Dynamic Workload Console supports a single connection to one Dynamic
Workload Broker engine at any given time for each authorized user. A different
connection is supported for each authorized user.

After having logged in to the Dynamic Workload Console using the administrator
user ID, or another user ID with assigned **TDWBAadministrator** or
**TDWBConfigurator** roles, follow these steps to create an engine connection to a
supported Dynamic Workload Broker engine:

1. In the Dynamic Workload Console, click **Dynamic Workload Broker** to expand
the tree.
2. Select **Configuration**.
3. Click **Server connection**.
4. In the Server Connection specify:
   - **Hostname**
     The host name of the Dynamic Workload Broker you want to connect
to.
   - **Non secure port**
     The non-secure port to be used for connection.
   - **Secure port**
     The secure port to be used for connection.
   - **Use Secure Connection**
     Specify whether a secure connection must be used. For more
information about security, see the *Tivoli Workload Scheduler:
Administration Guide, SC23-9113*.
   - **Username**
     Optionally specify a different user for the server connection. The
connection to the new server is enabled using the credentials of the
user you specified. Each user has access to only one server connection.
   - **Password**
     Specify the password for the authenticated user the connection applies
to.
5. Click **OK** to save your changes. The server connection you specified is enabled
and is immediately effective.

**Starting and stopping the Dynamic Workload Console**

To start and stop the Dynamic Workload Console or an engine, you must start and
stop the application server instance it is installed on.

**Embedded WebSphere Application Server on a Tivoli Integrated Portal in a
Tivoli Workload Automation instance**
If you installed the Dynamic Workload Console on the embedded WebSphere Application Server, you can start and stop the server as follows:

**Windows operating system:**
- To stop: `install_dir\wastools\stopWas.bat`
- To start: `install_dir\wastools\startWas.bat`

**UNIX and Linux operating systems:**
- To stop: `install_dir/wastools/stopWas.sh`
- To start: `install_dir/wastools/startWas.sh`

**WebSphere Application Server on a Tivoli Integrated Portal outside a Tivoli Workload Automation instance**

If you are using an external instance of WebSphere Application Server, use the following WebSphere Application Server scripts to start and stop an application server instance.

**Note:** These scripts can also be used to start and stop embedded WebSphere Application Server, although it is suggested that you use the method described above.

**Windows operating system:**
```
ewas_install_dir\bin\stopServer.bat app_server
  -user user_id -password user_id_pw
```
```
ewas_install_dir\bin\startServer.bat app_server
```

**UNIX and Linux operating systems:**
```
newas_install_dir/bin/stopServer.sh app_server
  -user user_id -password user_id_pw
```
```
/ewas_install_dir/bin/startServer.sh app_server
```

where:

**ewas_install_dir**
Is the directory where the WebSphere Application Server is installed.

**app_server**
Is the server name specified in the Tivoli Integrated Portal profile related to the Dynamic Workload Console or to the engine. The default is `server1`.

**user_id**
Is the administrator user ID specified when installing the Dynamic Workload Console or the engine.

**user_id_pw**
Is the administrator user ID password specified when installing the Dynamic Workload Console or the engine.
Configuring

This chapter describes how to configure the Dynamic Workload Console. You can perform the following optional configuration steps at any time after the installation.

- Configuring new users to access the Dynamic Workload Console
- Configuring the Dynamic Workload Console to use a user registry
  - Configuring the Dynamic Workload Console with LDAP - RACF (for more information, see WebSphere documentation at: Configuring to secure Lightweight Directory Access Protocol user registry using Resource Access Control Facility based on z/OS
- Configuring roles to access the Dynamic Workload Console
- Configuring the Dynamic Workload Console to use Single Sign-On
- Securing your communication using Secure Socket Layer protocol
- Configure the Dynamic Workload Console to launch in context

Note: If, after installing, you have more than one instance of WebSphere Application Server managing any Tivoli Workload Automation products, you must ensure that they have the same LTPA token_keys.

For information about configuration, see "Configuring the Dynamic Workload Console" in the Tivoli Workload Scheduler: Administration Guide, SC23-9113.

Getting started

This chapter explains how to get started with using the Dynamic Workload Console.

When you connect to the Dynamic Workload Console, you see a portfolio on the left with an entry for each Tivoli product installed inside the Tivoli Integrated Portal, such as, Tivoli Workload Scheduler.

You can access the Dynamic Workload Console from any computer in your environment using a web browser through both the secure HTTPS or HTTP protocol.

For an interactive overview of the product and its features, you can view several demo scenarios, available (in English only) on the product information center at the following link: https://www.ibm.com/developerworks/wikis/display/tivolimediagallery/Tivoli+Workload+Scheduler

The Dynamic Workload Console interface consists of the following sections:

Portfolio
   Located on the left, it has a tree structure and contains all the entries to launch the Dynamic Workload Console functions. Use the portfolio to navigate to the panels.

   Note: The upgrade does not report the customization you have performed on the portfolio. The defaults settings are installed.

Portlet area
   Your working area. It displays the panels corresponding to your selection in the portfolio. From each panel you can access the online help by clicking on the “?” symbol at the top right corner of the portlet.
Task bar
Contains a tab to open each active function you called from the portfolio. Each time you click an entry of the portfolio, the corresponding panel is opened in the portlet area. When you open a new panel, the preceding ones are minimized to tabs on the task bar and you can switch between the panels by clicking on these tabs. The browser task bar contains up to five open tabs. If you open more than five tabs, a new browser window opens and you can move from one page to another by opening the Select Action menu.

The portfolio has separate sections for Tivoli Workload Scheduler and Dynamic Workload Broker.

Tivoli Workload Scheduler portfolio
The Tivoli Workload Scheduler portfolio contains the following entries:

Quick Start
Open this entry to run some basic operations. Click here to create and manage queries of objects on the plan and to create and modify connections to the Dynamic Workload Console engines.

All Configured Tasks
Open this entry to view a list of all your saved tasks to monitor objects in the plan. A set of predefined tasks is provided to help you start using the application for the first time. These tasks cover the most common queries you might want to launch to find information about scheduling objects running on distributed, z/OS, or both platforms.

All Configured Reports
Open this entry to view a list of all your saved reports. From this view you can create new reports and customize existing ones.

Dashboard
Open this entry to have a graphical view that shows the progress of the current plan on the engines for which you configured a connection and specified its inclusion in the dashboard.

Workload
Manage your workload to design objects in the database, to handle plans, to submit jobs or job streams to monitor objects in the plan, or to generate reports.

Design
Open this entry to create, list, and edit object and object definitions in the database. Click here, for example, to create and modify jobs, job streams, and event rules.

Forecast
Open this entry to work with plans, creating and viewing trial and forecast plans and listing archived plans.

Submit
Open this entry to submit jobs and job streams on request

Monitor
Open this entry to create, list, and edit tasks to monitor objects in the plan. Click here, for example, to create and modify queries for jobs or job streams in the plan. Also, click here to handle queries about workload dependencies and events.
Scheduling Environment
Design and control the topology of your scheduling environment: the
workstations and domains.

**Design**
Open this entry to create, list, and edit workstations and domains
in your environment.

**Monitor**
Open this entry to create, list, and edit tasks to monitor
workstations and domains in the plan.

**Reporting**
Define and run reports.

**Generate Historical Reports**
Open this entry to create reports that gather historical data.

**Generate Plan Reports**
Open this entry to create reports with details about your plans.

**Generate Custom SQL Reports**
Open this entry to generate and run customized SQL reports.

**Settings**
Configure and modify general settings about Tivoli Workload Scheduler

**Manage Engines**
Open this entry to create, list, and edit your connections to the
Tivoli Workload Scheduler engine.

**Manage User Preferences**
Open this entry to configure and modify settings about table
layout, time zone, and dashboard layout for Tivoli Workload
Scheduler.

**Manage Settings**
Open this entry to import and export user preferences, configured tasks,
and engine connections and to configure your settings repository.

**Dynamic workload broker portfolio**
The Dynamic Workload Broker portfolio contains the following entries:

**Scheduling Environment**
Define and control logical resources and resource groups in your dynamic
scheduling environment

**Define New Logical Resource**
Open this entry to define a new logical resource required to run
jobs dynamically.

**Define New Resource Group**
Open this entry to create a new group definition to aggregate
different logical resources in a group.

**Logical Resources**
Open this entry to list and edit defined logical resources.

**Resource Groups**
Open this entry to list and edit defined resource groups

**Configuration**
Define a connection to the dynamic workload broker component.
Server Connections
Open this entry to create or edit a connection to the dynamic workload broker component.

Definitions
Manage dynamic workload to create list and submit jobs.

Define a New Job
Open this entry to create new dynamic job definitions.

Jobs
Open this entry to list, edit and submit dynamic workload objects.

Tracking
Monitor your dynamic workflow and the environment status.

Job Instances
Open this entry to monitor submitted dynamic job instances, see job output, and cancel jobs.

Computers
Open this entry to monitor and edit status and details of dynamic workstations.

Preferences
Customize display settings for Dynamic Workload Broker.

User Preferences
Open this entry to customize number of rows in each table page and set the displayed time zone.

First actions
The following sections describe the first and main actions you perform when you connect to the Dynamic Workload Console.

Creating a connection to a Tivoli Workload Scheduler engine
You type the details (such as IP address, user name, and password) to access a Tivoli Workload Scheduler engine, and, optionally, a database to operate with objects defined in plans or stored in the database. From the Dynamic Workload Console you can access the current plan, a trial plan, a forecast plan, or an archived plan for the distributed environment or the current plan for the z/OS® environment. You might want to access the database to perform actions against objects stored in it or generate reports showing historical or statistical data. In addition, working both on the database and on plans, you can create and run event rules to define and trigger actions that you want to run in response to events occurring on Tivoli Workload Scheduler nodes.

Defining a scheduling environment
You define your Tivoli Workload Scheduler network. You create workstation definitions on the database representing the physical machines or computer systems on which your workload is scheduled to run. Tivoli Workload Scheduler network is made up of the workstations where job and job stream processing occurs. When you design your network, you assign roles to these workstations to suit your specific business requirements. You can design your network with multiple domains, to divide control of a large network into smaller manageable groups. A typical Tivoli Workload Scheduler network consists of a workstation acting as master domain manager and at least one domain.

Defining scheduling objects in the database
You define your workload, which consists of jobs that are concatenated in
job streams. Then, you specify the calendars and run cycles according to which job streams must run. Moreover, you define possible dependencies to condition the workload processing. All these definitions can be done within the Workload Designer.

**Creating tasks to manage Tivoli Workload Scheduler objects in the plan**

You specify some filtering criteria to query a list of scheduling objects whose attributes satisfy the criteria you specified. Starting from this list, you can navigate and modify the content of the plan, switching between objects, opening more lists, and accessing other plans or other Tivoli Workload Scheduler environments.

**Creating a connection to a Tivoli dynamic workload broker scheduling environment**

You type the details (such as IP address, user name, password, and port) to access a dynamic workload broker workstation. Specify if you want to work in secure HTTPS or HTTP protocol. After creating the connection, opening the tracking computer you can view status and details of broker workstations, and define resources and dynamic jobs. For more details about dynamic scheduling, see: *Scheduling Workload Dynamically*.

---

**Upgrading**

This chapter describes how to upgrade the Dynamic Workload Console to the current version.

**Note:** The upgrade on a external WebSphere Application Server is not supported. If you have already installed the WebSphere Application Server with another product you must uninstall the Dynamic Workload Console and install it again.

To upgrade an instance of Tivoli Workload Automation:

- You must upgrade all components that are part of the instance. For example, if your instance includes one master domain manager and also the Dynamic Workload Console, you must upgrade both of these components.
- You must upgrade all the components part of that instance before you can upgrade the Dynamic Workload Console.

If you installed the Dynamic Workload Console sharing the WebSphere Application Server with other Tivoli Workload Automation components, when you upgrade those components, the existing Dynamic Workload Console will not work until you upgrade it to the new version.

When you upgrade the Dynamic Workload Console on the embedded WebSphere Application Server, changes are made to the directory structure of the console. Thus, the section contains the following topics:

- “Updating authentication”
- “Upgrading the console installed on an embedded WebSphere Application Server” on page 267
- “Performing the upgrade” on page 268

**Updating authentication**

This section describes how your configured authentication mechanism is upgraded.

---

[266] IBM Tivoli Workload Scheduler: Planning and Installation
In versions of Tivoli Workload Scheduler before V8.6, authentication was configured to use stand-alone user registries, managed by the embedded WebSphere Application Server. The available options were:

- Local operating system
- Custom (through PAM - Pluggable Authentication Module)
- LDAP
- File Registry

If you enabled LDAP, you could use one of the following servers:

- IBM Tivoli Directory Server
- Sun ONE
- Microsoft Windows Active Directory
- RACF configured on IBM Tivoli Directory Server

Versions of Tivoli Workload Scheduler from V8.6 onwards are configured for authentication (through the embedded WebSphere Application Server) in VMM (Virtual Member Manager) mode. This creates a Federated User Registry, which supports the simultaneous use of more than one user registry. The user registry choices and LDAP server options are similar to those in versions before V8.6.

During the upgrade, your existing configuration are migrated, so that when the upgrade is complete the product is configured to use the same authentication mechanism as before, but within a Federated User Registry.

For detailed information, see Tivoli Workload Scheduler: Administration Guide.

### Upgrading the console installed on an embedded WebSphere Application Server

This section provides information about the upgrade of the Dynamic Workload Console on an embedded WebSphere Application Server.

### Directory structure

This section describes the program directory structure and the directory structure for SSL files that was implemented in version 8.5.1. This section applies if you are upgrading from version 8.3 or 8.4. If you are upgrading from version 8.5 or 8.5.1, this directory structure already exists.

**Program directory:**

When you upgrade the Dynamic Workload Console to the current version, a new program directory structure is created. During the upgrade process, components of the Dynamic Workload Console are moved from the old directory structure and then updated into the new directory structure. The Dynamic Workload Console program files remain in the original installation directory.

If you have any custom configurations (for example, custom scripts or backup processes) existing in your Dynamic Workload Console structure, you must update them so that they work in the new directory structure.

For example, if you originally installed the Dynamic Workload Console into the default directory `c:\Program Files\IBM\webui\`, you have a directory structure as follows:
When you upgrade the Dynamic Workload Console, the new directory structure is:

```
c:\Program Files\IBM\TWA\eWAS
```

The new directory structure includes new embedded WebSphere Application Server tools that are common to Tivoli Workload Scheduler.

**Directory for SSL files:**
When you upgrade to the current version, a new directory for SSL files is created. The following describes the old and new directory structures.

**Note:** Before upgrading you must backup any customized SSL keys and copy them to the default installation path.

After upgrading, the old SSL files stored in PCKS12 format are imported into new SSL files in JKS format.

The old PCKS12 files are also copied to the new directory as a backup. The `key.p12` file becomes `TWSServerKeyFile.jks`. The `trust.p12` files becomes `TWSServerTrustFile.jks`.

**Previous directory structure**
- `TWSSInstallationPath\appServer\profiles\webuiprofile\config\cells\CellName\nodes\ModuleName\key.p12`
- `TWSSInstallationPath\appServer\profiles\webuiprofile\config\cells\CellName\nodes\ModuleName\trust.p12`

**New directory structure**
- `TWSSInstallationPath\eWAS\profiles\TIPProfile\config\cells\TIPNode\nodes\TIPCell\key.p12`
- `TWSSInstallationPath\eWAS\profiles\TIPProfile\config\cells\TIPNode\nodes\TIPCell\trust.p12`
- `TWSSInstallationPath\eWAS\profiles\TIPProfile\etc\TWSServerKeyFile.jks`
- `TWSSInstallationPath\eWAS\profiles\TIPProfile\etc\TWSServerTrustFile.jks`

**Note:** The files `key.p12` and `trust.p12` are not used by the Dynamic Workload Console, but are backed up.

**Performing the upgrade**
You can upgrade the Dynamic Workload Console using the following methods:

**Interactive wizard**
To upgrade using the interactive wizard, run the setup for the operating system on which you are installing:
On Windows operating system:

`WINDOWS\SETUP.exe`

Before beginning, stop the `appserverman` process by running the following commands:

`Shutdown.cmd -appsrv`
`StartWas.bat -direct`

After running these commands, verify that all Tivoli Workload Scheduler processes are stopped with the exception of the embedded WebSphere Application Server. The embedded WebSphere Application Server must remain running.

On UNIX and Linux operating systems:

`SETUP.sh` or `operating_system/SETUP.bin`.

Note: SETUP.sh copies the entire image to a temporary directory. Ensure there is enough space available.

Launchpad
Start the launchpad and select the Dynamic Workload Console upgrade. The installation wizard is launched with some options pre-selected to upgrade the console.

Silent
You can run the upgrade silently and in background by creating a response file from the template provided and running the installation wizard with the `-silent` option. See “Performing a silent installation” on page 223 for more details on how to run a silent installation or upgrade. See the following section for the upgrade information that you must supply in the response file.

Follow the installation panels to complete the upgrade. The following list describes the fields you must provide during the upgrade.

Use an existing instance of the Dynamic Workload Console
When you are prompted that a previous version of the Dynamic Workload Console has been found, select Use an existing instance. From the drop-down list, choose the instance that you are upgrading.

Administrative credentials of application server
Enter the external or embedded WebSphere Application Server user name and password.

Backup directory
Choose a backup directory. This directory contains only configuration information and other program-related objects and not the external or embedded WebSphere Application Server files. Note that this directory remains on your computer even after the upgrade is complete.

IPC connector
The IPC connector on the portal. The default value is 29314.

REST notification
The REST notification port on the portal. The default value is 29324.

DCS Unicast port
The DCS Unicast port on the portal. The default value is 29353.

Note:

1. For information about Tivoli Workload Automation instances, see “Instances of Tivoli Workload Automation” on page 247.
2. During an upgrade on Windows, the embedded WebSphere Application Server Windows Service account name in the local OS user registry is changed to the administrator user name of the Tivoli Integrated Portal. If you use a custom registry or LDAP registry, the service is upgraded to the installation user.

3. It is not necessary to manually stop the embedded WebSphere Application Server prior to upgrading, as it is stopped automatically during the upgrade procedure.

Uninstalling

This chapter describes how to uninstall the Dynamic Workload Console. It is divided into the following sections:

- “Uninstalling using the wizard”
- “Uninstalling in silent mode”

Uninstalling using the wizard

To uninstall the Dynamic Workload Console using the wizard, perform the following steps:

1. Start the Tivoli Integrated Portal.
2. Start the uninstall as follows:

   **On Windows operating system:**
   Perform one of the following:
   - From the `TWA_home\TDWC` directory, run the command:
     `uninstall.bat`
   - From the Control Panel, click **Add/Remove Programs**. Scroll down the list of software, and select the Dynamic Workload Console. Click **Change/Remove**.

   **On UNIX operating systems:**
   From the `TWA_home/TDWC` directory, run the command:
   `uninstall.sh`

3. Select the language.
4. Click **Next** in the Dynamic Workload Console uninstall welcome window.
5. Provide the external or embedded WebSphere Application Server administrator user name and password, and click **Next**.
6. In the uninstall summary window, check that the directory from where the product is to be removed and the features to be removed are correct, and then click **Uninstall**. If you installed the Dynamic Workload Console and the Tivoli Integrated Portal, they are both uninstalled. If you installed the Dynamic Workload Console on a existing Tivoli Integrated Portal only the Dynamic Workload Console is uninstalled.
7. When the uninstall completes, a window showing a message about the success of the operation is displayed. Click **Finish** to exit the InstallShield Wizard.

Uninstalling in silent mode

You can perform a silent uninstall of the Dynamic Workload Console.

Before starting to uninstall ensure that the Tivoli Integrated Portal is active, and move to a directory different from the `tdwc_install_dir`.

Run the uninstall command as follows:
On Windows operating system:
```
twa_home\tdwc\uninstall.bat -options
response_file.txt -silent
```

On UNIX or Linux operating systems:
```
twa_home/tdwc/uninstall.bin -options
response_file.txt -silent
```

where `response_file` is the full path name.

---

**Troubleshooting the installation, upgrade, and uninstallation**

This chapter describes how to troubleshoot the installation, upgrade, and uninstallation of the Dynamic Workload Console. It is divided into the following sections:

- “Installation and uninstallation log and trace files”
- “Recovering a failed InstallShield wizard installation”
- “Recovering a failed upgrade” on page 272
- “Manually uninstall an integrated Dynamic Workload Console” on page 272
- “Troubleshooting scenarios” on page 275

**Note:** See the section “Manually uninstall an integrated Dynamic Workload Console” on page 272 to manually uninstall or recover from a failed installation.

---

**Installation and uninstallation log and trace files**

For information about installation log files, see “Installation log files” on page 248.

**Recovering a failed InstallShield wizard installation**

The recovery of a failed installation of the Dynamic Workload Console is structurally very similar to that described in *Tivoli Workload Scheduler: Planning and Installation Guide* and for Tivoli Workload Scheduler. However, there are some significant differences, which are described in this section.

The recovery of a failed installation is fully described in the *Tivoli Workload Scheduler: Planning and Installation Guide*.

Follow the instructions in *Tivoli Workload Scheduler: Planning and Installation Guide*

Follow the instructions in the *Tivoli Workload Scheduler: Planning and Installation Guide* up to the point where you want to modify the values of a step, and then follow these instructions:

1. The values used in each step for the Dynamic Workload Console are all stored in one place - *Step 0*. So if you discover, for example, that the step that configures the embedded Tivoli Integrated Portal has failed because a port is in use, you must go to *Step 0* and modify the value for the port in that step.
2. Set the status of *Step 0*, plus the status of the step that failed, to *Ready*.
3. In all cases, run *Step 0* in the Step List, using the *Run next* option. *Step 0* uses the original data, as modified by you, to regenerate all of the scripts that run the steps.
4. Resume the wizard from the failed step, either running *Run all* to complete the installation without stopping at each step, or *Run next*, to complete the installation step by step.
Note: You cannot rerun any step that has completed successfully, other than Step 0.

Recovering a failed upgrade

In the case of a failed upgrade, contact IBM Software Support.

Manually uninstall an integrated Dynamic Workload Console

Perform the following steps to manually remove an instance of Tivoli Workload Automation that contains the Dynamic Workload Console and uses the embedded Tivoli Integrated Portal. In the case of a failed installation, you may find some of the steps are unnecessary, depending on when the installation failed.

To remove an instance of Tivoli Workload Automation that contains an integrated installation of Tivoli Workload Scheduler and the Dynamic Workload Console, do the following actions:

1. Uninstall Tivoli Workload Scheduler as described in the Tivoli Workload Scheduler: Planning and Installation Guide.
2. Remove the Dynamic Workload Console by performing the steps described in the procedure below.

If you want to remove the Dynamic Workload Console from an instance of Tivoli Workload Automation without removing the Tivoli Workload Automation instance, contact IBM Software Support.

Note: Only perform these manual steps on systems where the Dynamic Workload Console is installed, otherwise you delete the Composite Offering Installer registry.

On Windows operating system:

1. If you have already removed the Dynamic Workload Console, using the installation DVD or the downloaded elmages, run the following command from the \TDWC\WEBUI\<operating_system>\scripts\ directory:
   
   
   # ./cleanDE.bat -installRoot <TWA_INSTALL_DIR>\eWas -force true

   If, instead, the Dynamic Workload Console is still installed but not working, run the following command from the <TWA_INSTALL_DIR>\TDWC\tdwcutils\scripts\ directory:
   
   # ./cleanDE.bat -installRoot <TWA_INSTALL_DIR>\eWas -force true

2. Stop the service:
   
   <TWA_INSTALL_DIR>\bin\WASService -stop TIPProfile_PortDefaultValuePort

3. Remove the service:
   
   <TWA_INSTALL_DIR>\bin\WASService -remove

4. Navigate to the <TWA_INSTALL_DIR> and note the name of the ID file twainstancexxx.id. You will need this information later in the procedure.

5. Remove the directory:
   
   <TWA_INSTALL_DIR>

6. Remove the directory:
   
   C:\Program Files\Common Files\InstallShield\Universal\TDWC

7. Delete the following registry key:

   HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\e625666383dedb70850864e2a6feaa2e1371705039
8. Remove the file:
   %windir%\TWA\twainstancexxx.properties
9. Restart the system.

On UNIX and Linux operating systems:
1. If you have already removed the Dynamic Workload Console, using the installation DVD or the downloaded eImages, run the following command from the \TDWC\WEBUI\<operating_system>\scripts\ directory:
   # ./cleanDE.sh -installRoot <TWA_INSTALL_DIR>/eWas -force true
   
   If, instead, the Dynamic Workload Console is still installed but not working, run the following command from the <TWA_INSTALL_DIR>/TDWC\_tdwcutils\scripts\ directory:
   # ./cleanDE.sh -installRoot <TWA_INSTALL_DIR>/eWas -force true
2. Stop the server by running the command:
   <TWA_INSTALL_DIR>/wastools/stopWas.sh
3. Navigate to the <TWA_INSTALL_DIR> and note the name of the ID file twainstancexxx.id. You will need this information later in the procedure.
4. Remove the directory:
   <TWA_INSTALL_DIR>
5. Remove the directory:

   On AIX
   /usr/lib/objrepos/InstallShield/Universal/TDWC

   On all UNIX systems, except AIX
   ROOT_USER_HOME/InstallShield/Universal/TDWC
6. Remove the file:
   etc/TWA/twainstancexxx.properties

Manually uninstall a stand-alone Dynamic Workload Console version 8.6.0 instance

The correct procedure to uninstall a stand-alone Dynamic Workload Console instance is to run its uninstaller program. If this is not possible, for example for a failed installation or uninstallation, a manual procedure is required.

Perform the following steps to manually remove a stand-alone instance of Dynamic Workload Console that uses the embedded Tivoli Integrated Portal. In the case of a failed installation, you may find some of the steps are unnecessary, depending on when the installation failed.

To remove an instance of Dynamic Workload Console, perform the following actions.

Note: Only perform these manual steps on systems where the Dynamic Workload Console is installed, otherwise you delete the Composite Offering Installer registry.

On Windows operating system:
1. If you have already removed the Dynamic Workload Console, using the installation DVD or the downloaded eImages, run the following command from the \TDWC\WEBUI\<operating_system>\scripts\ directory:
# ./cleanDE.bat -installRoot \TWA_INSTALL_DIR\eWas -force true

If, instead, the Dynamic Workload Console is still installed but not working, run the following command from the <TWA_INSTALL_DIR>\TDWC\tdwutils\scripts\ directory:
# ./cleanDE.bat -installRoot \TWA_INSTALL_DIR\eWas -force true

2. Stop the service:
<TWA_INSTALL_DIR>\bin\WASService -stop TIPProfile_Port_defaulthost_port

3. Remove the service:
<TWA_INSTALL_DIR>\eWAS\bin\wasservice.exe -remove TDWC_admin_username

4. Remove the ISMP installation registry directory:
%CommonProgramFiles%\InstallShield\Universal\TDWC

5. Remove the Tivoli Workload Automation registry file. In the file <TWA_INSTALL_DIR>\twainstancexxx.id, take note of the number n, which indicates the Tivoli Workload Automation instance number, and remove the following files:
%WINDIR%\TWA\twainstance<n>.TWA.properties
%WINDIR%\TWA\twainstance<n>.TWA.properties.ext

6. Delete the following registry key:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\ 6256d6383dedb7030864e2a6feaa2e1371705039

7. Remove the directory:
<TWA_INSTALL_DIR>

On UNIX and Linux operating systems:

1. If you have already removed the Dynamic Workload Console, using the installation DVD or the downloaded eImages, run the following command from the \TDWC\WEBUI<operating_system>\scripts\ directory:
# ./cleanDE.sh -installRoot \TWA_INSTALL_DIR\eWas -force true

If, instead, the Dynamic Workload Console is still installed but not working, run the following command from the <TWA_INSTALL_DIR>\TDWC\tdwutils\scripts\ directory:
# ./cleanDE.sh -installRoot \TWA_INSTALL_DIR\eWas -force true

2. Stop the Embedded WebSphere Application Server by running the command:
<TWA_INSTALL_DIR>/wastools/stopWas.sh

3. Remove the directory:

On AIX
˝/usr/lib/objrepos/InstallShield/Universal/TDWC

On all UNIX systems, except AIX
RESET_USER_HOME/InstallShield/Universal/TDWC

4. Remove the Tivoli Workload Automation registry file. In the file <TWA_INSTALL_DIR>\twainstancexxx.id, take note of the number n, which indicates the Tivoli Workload Automation instance number, and remove the following files:
/etc/TWA/twainstance<n>.TWA.properties
/etc/TWA/twainstance<n>.TWA.properties.ext

5. Remove the directory:
6. Remove the following directories:
   /var.ibm/common/acsi
   /usr.ibm/common/acsi

Troubleshooting scenarios

The troubleshooting scenarios are listed in the following categories:

- “Problems with the launchpad”
- “Problems with the interactive wizard”
- “Problems with the silent installation” on page 279
- “Problems with the upgrade” on page 280
- “Problems with the uninstallation” on page 280

Problems with the launchpad

The following problems might be encountered while using the launchpad to install the Dynamic Workload Console:

- “Warning messages displayed when using the launchpad on Linux”
- “Undefined error when using launchpad on Windows operating system”

Warning messages displayed when using the launchpad on Linux:

Problem description:

Warning messages might be displayed on the standard output when using the launchpad on Linux.

Cause and solution

You can ignore these messages because they do not indicate a malfunction of the launchpad.

Undefined error when using launchpad on Windows operating system:

Problem description:

You try to install the Dynamic Workload Console on a Windows operating system using the launchpad and you get an "Undefined" error message. The launchpad does not start.

Cause and solution

Make sure that the path from where you launched the installation does not contain folder names longer than eight characters. If it does, then map the path to the launchpad.exe, and run the launchpad from that new path.

Problems with the interactive wizard

The following problems might be encountered while running the Dynamic Workload Console interactive installation:

- “The Dynamic Workload Console installation hangs” on page 276
- “Installation hangs during stopWas command” on page 276
- “Tivoli Integrated Portal installation fails even if into the logs you find successfully installed” on page 276
- “Installation from a remote shared folder fails on Windows operating system” on page 277
The Dynamic Workload Console installation hangs:

Problem description:

The installation of the Dynamic Workload Console does not proceed. This occurs regardless of the method you used to install.

Cause and solution

Make sure an active personal firewall is not preventing the installation process from connecting to the network. If it is, allow the connection and then continue with the installation.

Installation hangs during stopWas command:

Problem description:

The installation of the Dynamic Workload Console does not proceed.

Cause and solution

The installation of the Dynamic Workload Console does not proceed because the stopWas command is hanging.

To continue the installation, open the Task Manager and locate the Java process of the embedded WebSphere Application Server. This process is the java.exe process with the associated installation username. Stop this process.

Then, find the associated WASService process. This process is the WASService.exe process with the associated the installation username. Stop this process.

Continue with the installation.

Tivoli Integrated Portal installation fails even if into the logs you find successfully installed:

Problem description:

You are installing the Tivoli Integrated Portal, either using the wizard or the silent installation, and the installation fails with the following error:

Start installing TIP - Tivoli Integrated Portal 2.2
For log details, please see
C:\Documents and Settings\Administrator\TIPInstaller-00.log and {install location}\logs.zip
Preparing SILENT Mode Installation...

*******************************************************************************
GenericInstaller (created with InstallAnywhere by Macrovision)
Installing...

Installation Complete.
SUCCESS: The overall installation is successful.
Current OS is Windows XP

Executing 'C:\IBM\TWS\TDWC\eWAS\profiles\TIPProfile\bin\tipcli.bat' with arguments: 'Version'

The ' characters around the executable and arguments are not part of the command.

Execute:Java13CommandLauncher:

Executing 'C:\IBM\TWS\TDWC\eWAS\profiles\TIPProfile\bin\tipcli.bat' with arguments: 'Version'

The ' characters around the executable and arguments are not part of the command.

The following error occurred while executing this line:
C:\temp\TDWC\WEBUI\WINDOWS\xml\commonTargets.xml:1735:
Execute failed: java.io.IOException:
Cannot run program "C:\IBM\TWS\TDWC\eWAS\profiles\TIPProfile\bin\tipcli.bat"
in directory "C:\IBM\TWS\TDWC\eWAS\profiles\TIPProfile\bin":
CreateProcess error=2, The system cannot find the file specified.

Installation Complete.
SUCCESS: The overall installation is successful.
Current OS is Windows XP

Executing 'C:\IBM\TWS\TDWC\eWAS\profiles\TIPProfile\bin\tipcli.bat' with arguments: 'Version'

The ' characters around the executable and arguments are not part of the command.

Execute:Java13CommandLauncher:

Executing 'C:\IBM\TWS\TDWC\eWAS\profiles\TIPProfile\bin\tipcli.bat' with arguments: 'Version'

Cause and solution

This error occurs if you remove an instance of the Dynamic Workload Console, and the Tivoli Integrated Portal is also installed in the same path, without running the uninstaller (for example, removing manually only the content of the TWA directory). In this case, the Tivoli Integrated Portal instance remains registered in the Deployment Engine installation registries and when you perform another installation of the Tivoli Integrated Portal the installer reports the error indicated previously.

Run the procedure described in “Manually uninstall an integrated Dynamic Workload Console” on page 272 to clean your environment. Restart the installation from the failed step.

Installation from a remote shared folder fails on Windows operating system:

Problem description:

You try to install the Dynamic Workload Console on a Windows operating system from a shared network folder that uses Universal Naming Convention (UNC). The installation fails.

Cause and solution
You must map the remote folder locally on the Windows system where you want to install the Dynamic Workload Console and then run the installation using the local path.

Installation fails on a Linux 390 system with a hostname which is not a Fully Qualified Domain Name:

Problem description:

You install the Dynamic Workload Console with the embedded WebSphere Application Server on a server with a hostname is not a Fully Qualified Domain Name. The installation fails and the following error is stored in the twainstall.log file:

`ADMU3011E: Server launched but failed initialization. startServer.log, SystemOut.log(or job log in zOS) and other log files under /oracle/ibm/TDWC/eWAS/profiles/TIPProfile/logs/tdwcserver should contain failure information.`

`WASX7023E: Error creating "SOAP" connection to host "localhost"; exception information:
com.ibm.websphere.management.exception.ConnectorNotAvailableException:
com.ibm.websphere.management.exception.ConnectorNotAvailableException:
ADMC0016E: The system cannot create a SOAP connector to connect to host localhost at port 28880.`

Cause and solution

Run the following command from the system prompt on the Linux 390 system where you tried to install the Dynamic Workload Console:

`hostname --fqdn`

If the command returns:

`hostname: Unknown host`

the host name is not resolved. You must specify a hostname with a fully qualified domain name to install the embedded WebSphere Application Server. Update the hostname notation, as explained in the embedded WebSphere Application Server documentation, and then rerun the installation.

Java Virtual Machine (JVM) failure when installing the Dynamic Workload Console on a Red Hat Enterprise Linux (RHEL) Version 5 or a Suse Linux system Version 11:

Problem description:

When installing the Dynamic Workload Console on a Red Hat Enterprise Linux Version 5 or a Suse Linux Version 11 system, you might receive the error "Failed to find VM - aborting".

Cause and solution

Linux systems have a security feature named 'Security Enhanced Linux', or SELinux for short. A weaker version of SELinux was included in Red Hat Enterprise Linux Version 4, and was disabled by default. On these versions of Red Hat Enterprise Linux and Suse Linux, this security feature is enabled by default. SELinux helps to keep the host secure from certain types of malicious attacks. However, the default settings have been known in many cases to prevent Java from running properly.
To fix this issue, choose one of the following options:

- Configure SELinux so that it knows that the Dynamic Workload Console Java related processes are acceptable to run.
- Change the mode of SELinux to *Permisive* by entering `setenforce 0` on the command line. SELinux will be fully enabled again the next time the system is rebooted or if `setenforce 1` is entered on the command line. For the Dynamic Workload Console to function, you must set `setenforce 0`. For more information about `setenforce`, see the documentation for your operating system.

### The Dynamic Workload Console graphical installation and uninstallation fail to start on Red Hat Enterprise Linux (RHEL) Version 5 on x86-64:

#### Problem description:

When launching the Dynamic Workload Console installation or uninstallation wizard in interactive mode on Red Hat Enterprise Linux (RHEL) Version 5 x86-64, you might receive the following error:

For the installation:

The installer is unable to run in graphical mode.
Try running the installer with the -console or -silent flag.

For the uninstallation:

The uninstaller is unable to run in graphical mode.
Try running the uninstaller with the -console or -silent flag.

#### Cause and solution

If you run into this problem, launch the installation or the uninstallation in silent mode. For more information, see “Performing a silent installation” on page 223 and “Uninstalling in silent mode” on page 270.

### On Windows, the Dynamic Workload Console installation fails if you try to reinstall on a different profile of an external WebSphere Application Server:

#### Problem description:

This situation applies to a Windows operating system. You install the Dynamic Workload Console on a profile, for example, ProfileA, of an existing WebSphere Application Server installation. You remove the Dynamic Workload Console successfully and then try to install it on a different profile of the same WebSphere Application Server. The installation fails.

#### Cause and solution

A possible cause is that when you removed the Dynamic Workload Console, some files belonging to ProfileA were not removed. To solve this problem, stop ProfileA and then install the Dynamic Workload Console again on the other profile.

### Problems with the silent installation

The following problems might be encountered while running the Dynamic Workload Console silent installation:

- “The silent uninstallation does not work and an error code is returned” on page 280
- “Tivoli Integrated Portal installation fails even if into the logs you find successfully installed” on page 276
The silent uninstallation does not work and an error code is returned:

**Problem description:**

If you try to perform a silent uninstall with a response file that does not exist, either because the file name is incorrect or because you specified the wrong directory, an error code is returned and the uninstallation does not run. Nothing is logged in the temporary directory and no messages are issued.

**Cause and solution**

Ensure that you specify a valid response file name.

**Problems with the upgrade**

The following problem might be encountered while running the Dynamic Workload Console upgrade:

- "Upgrade fails with message AWSUI0085E"

**Upgrade fails with message AWSUI0085E:**

**Problem description:**

You are running the upgrade of the Dynamic Workload Console and the following error occurs:

The instance of the Dynamic Workload Console that you want to upgrade uses an LDAP user registry.

The LDAP server type you are using is not supported.

The supported LDAP server types are:


Use a supported LDAP server type or install a fresh instance of the Dynamic Workload Console

**Cause and solution**

The upgrade fails because you are using an LDAP server type that is not supported. The supported LDAP server types are:

- IBM Tivoli Directory Server
- Microsoft Active Directory
- Sun ONE DS
- RACF configured on IBM Tivoli Directory Server

Configure the Dynamic Workload Console to use:

- One of the supported LDAP server types.
- The local operating system - the default authentication system at installation on Windows operating systems.
- The custom (through PAM - Pluggable Authentication Module) - the default authentication system at installation on UNIX and Linux operating systems.

or install a fresh instance of the Dynamic Workload Console and then configure it to use one of the supported LDAP server types.

**Problems with the uninstallation**

The following problems might be encountered while running the Dynamic Workload Console uninstallation:

- "Uninstall fails on Windows if the installation directory contains the @ character" on page 281
- "The Dynamic Workload Console interactive uninstallation wizard fails to start on Red Hat Enterprise Linux (RHEL) Version 5 on x86-64" on page 281
“Installation fails when reinstalling the Dynamic Workload Console after having
uninstalled it”

Uninstall fails on Windows if the installation directory contains the @
character:
Problem description:

When running uninstaller.exe to remove the Dynamic Workload Console installed
in a directory that contains the @ character, for example C:\Program
Files\ibm\TDWC\a-@a, the uninstall fails and the following error message is
displayed:
CreateProcess failed ==> The system cannot find the file specified.

Cause and solution

The uninstall fails because '@' is a special character for ISMP. ISMP is not able to
manage an installation directory containing this character.

You can bypass the problem by running the uninstall as follows:
"C:\Program Files\ibm\TDWC\a-@a\_jvm\bin\java.exe"
- cp "C:\Pr Fi\ibm\TDWC\a-@a\tdwuninst\uninstall.jar" run

Run this command outside the installation directory, otherwise the installation
directory is not removed.

The Dynamic Workload Console interactive uninstallation wizard fails to start
on Red Hat Enterprise Linux (RHEL) Version 5 on x86-64:
Problem description:

When launching the Dynamic Workload Console interactive uninstallation wizard
on Red Hat Enterprise Linux (RHEL) Version 5 x86-64, the following error is
displayed:
The uninstaller is unable to run in graphical mode.
Try running the uninstaller with the -console or -silent flag.

See “The Dynamic Workload Console graphical installation and uninstallation fail
to start on Red Hat Enterprise Linux (RHEL) Version 5 on x86-64” on page 279 for
the solution.

Installation fails when reinstalling the Dynamic Workload Console after having
uninstalled it:
Problem description:

Installation fails when trying to reinstall the Dynamic Workload Console on a
Windows system where the Dynamic Workload Console has been uninstalled.

Cause and solution

This problem can be due to the fact that eWAS directory was not correctly deleted
during uninstallation. If during the Dynamic Workload Console uninstallation eWAS
directory cannot be deleted because it is locked by another process, the
uninstallation wizard does not fail but completes successfully without removing
the directory. The solution for this problem is to force the uninstallation to fail
when eWAS directory cannot be deleted. In this way you can kill all the processes
related to the eWAS directory. Alternatively, you can manually delete it and finally
rerun the installation step.
Part 4. Tutorials

Using the Tivoli Workload Scheduler tutorial utility

This section describes the Tivoli Workload Scheduler tutorial utility and guides you through a set of steps to populate and use a stand-alone test environment. The tutorial utility is intended for first-time users of Tivoli Workload Scheduler who want an overview of the features and capabilities of the product in a real environment. The tutorial utility includes a sample database with predefined scheduling objects and a set of scenarios that use these objects.

The `sampledbsetup.sh` or the `SAMPLEDBSETUP.CMD` script (depending on whether you are in a UNIX or Windows environment) populates your Tivoli Workload Scheduler with a set of scheduling objects. The scenario scripts use these objects in basic scheduling activities. Each scenario is self-contained and can be run in any order, with the exception of the first scenario which is a prerequisite to all others.

The Tivoli Workload Scheduler tutorial utility runs only on a master domain manager. It does not affect any other workstation defined in your Tivoli Workload Scheduler environment. Each scenario is launched as a separate script file which uses the `conman` and `composer` command interfaces. The syntax and usage of each command used in the scenarios is explained in detail in the Tivoli Workload Scheduler: User’s Guide and Reference. Before you begin using the utility, read an overview of Tivoli Workload Scheduler concepts and tasks in Tivoli Workload Automation: Overview.

This chapter is divided into the following sections:

- "Populating your Tivoli Workload Scheduler database" on page 285
- "Overview of the scheduling scenarios" on page 285
- "Creating and working with the production plan" on page 285
- "Running the scheduling scenarios" on page 286
- "Removing tutorial objects from the database" on page 290

Populating your Tivoli Workload Scheduler database

This section describes how you use the utility to populate your Tivoli Workload Scheduler database.

After you have installed Tivoli Workload Scheduler on the master domain manager in your test environment you are ready to populate the database.

Follow these steps:

1. Go to the `TWS_home/TWS/TWSTutorial` directory, where `TWS_home` is the home directory of the user for which you installed Tivoli Workload Scheduler.
2. Launch the tutorial utility installation script:
   - In a Windows operating system:
     `SAMPLEDBSETUP.CMD`
   - In a UNIX and Linux operating systems:
     `sampledbsetup.sh`
The script adds a set of scheduling objects with names starting with the string *
SMPL*, followed by the object type and scenario number so that all objects used in
each scenario are easily identifiable. Some objects are different depending on
whether you are using a UNIX or a Windows environment.

The script performs a check on the database. If any objects with the same name are
found, you are prompted to specify if these objects can be overwritten.

When processing of the script ends successfully, your Tivoli Workload Scheduler
database contains the objects needed to run the scheduling scenarios.

**Objects used by the Tivoli Workload Scheduler tutorial scenarios**

After you have successfully installed the Tivoli Workload Scheduler tutorial utility
in your test environment, your database is populated with the following
scheduling objects:

| Table 21. Objects downloaded by the tutorial utility |
|---------------------------------|---------------------------------|
| **Object type** | **Object Names (Total objects)** |
| Calendar | SMPCAL6 (1) |
| Variable | SMPLHOME, SMPLUSER, SMPLWIN1 to SMPLWIN4 or SMPLUNIX1 to SMPLUNIX4, SMPLSLEEP, SMPLTMP, SMPLPATH (6) |
| Resource | SMPLRES1, SMPLRES2 (2) |
| Prompt | SMPLPRM4, SMPLPRM5, SMPLPRM6, SMPLPRM7 (4) |
| Job | SMPL_JOB_3_0_1, SMPL_JOB_3_0_2, SMPL_JOB_3_0_3, SMPL_JOB_4_0_1, SMPL_JOB_4_0_2, SMPL_JOB_4_0_3, SMPL_JOB_5_0_1, SMPL_JOB_5_0_2, SMPL_JOB_7_0_1, SMPL_JOB_7_0_2, SMPL_JOB_7_0_3, SMPL_JOB_9_0_1, SMPL_JOB_9_1_1, SMPL_JOB_EVN, SMPL_JOB_ODD, SMPL_JOB_PAIR, SMPL_JOB_SBJ, SMPL_JOB_7_0_LAST, SMPL_JOB_7_0_RECV (19) |
| Job Stream | SMPL_SCHED_3_0_1, SMPL_SCHED_3_0_2, SMPL_SCHED_4_0_1, SMPL_SCHED_4_0_2, SMPL_SCHED_4_0_3, SMPL_SCHED_4_0_S, SMPL_SCHED_5_0_1, SMPL_SCHED_5_0_2, SMPL_SCHED_7_0_1, SMPL_SCHED_7_0_2, SMPL_SCHED_7_0_3, SMPL_SCHED_9_0_1, SMPL_SCHED_9_0_2, SMPL_SCHED_9_1_1, SMPL_SCHED_5–ODD, SMPL_SCHED_5_EVN, SMPL_SCHED_SBS (17) |
| Event Rule | SMPL_FILTER_RULE (1) |
| Variable table | SMPL_VAR_TABLE_9_0_1, SMPL_VAR_TABLE_9_0_2 (2) |

You can display each object by running the **composer** command interface. For
specific information about the syntax of the **composer** interface, see the *Tivoli
Overview of the scheduling scenarios

The following table describes the topics covered in each scenario. Each scenario is a separate script file.

You must run Scenario 1 first, but you can choose to run the other scenarios in any order.

Table 22. List of scheduling scenarios

<table>
<thead>
<tr>
<th>Scenario name</th>
<th>Script name</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario 1</td>
<td>scenario1.0.bat (Windows)</td>
<td>Creating the production plan and viewing its contents</td>
</tr>
<tr>
<td></td>
<td>scenario1.0.sh (UNIX)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: This scenario is a prerequisite for all the other scenarios in your sequence.</td>
</tr>
<tr>
<td>Scenario 2</td>
<td>scenario2.0.bat (Windows)</td>
<td>Administrative commands: starting and stopping Tivoli Workload Scheduler processes</td>
</tr>
<tr>
<td></td>
<td>scenario2.0.sh (UNIX)</td>
<td></td>
</tr>
<tr>
<td>Scenario 3</td>
<td>scenario3.0.bat (Windows)</td>
<td>Scheduling basics: how jobs are scheduled, run order of jobs</td>
</tr>
<tr>
<td></td>
<td>scenario3.0.sh (UNIX)</td>
<td></td>
</tr>
<tr>
<td>Scenario 4</td>
<td>scenario4.0.bat (Windows)</td>
<td>Advanced Scheduling: prompt, file, and resource dependencies</td>
</tr>
<tr>
<td></td>
<td>scenario4.0.sh (UNIX)</td>
<td></td>
</tr>
<tr>
<td>Scenario 5</td>
<td>scenario5.0.bat (Windows)</td>
<td>Time dependencies and run cycles</td>
</tr>
<tr>
<td></td>
<td>scenario5.0.sh (UNIX)</td>
<td></td>
</tr>
<tr>
<td>Scenario 6</td>
<td>scenario6.0.bat (Windows)</td>
<td>Job submission (jobs, job streams, ad-hoc jobs)</td>
</tr>
<tr>
<td></td>
<td>scenario6.0.sh (UNIX)</td>
<td></td>
</tr>
<tr>
<td>Scenario 7</td>
<td>scenario7.0.bat (Windows)</td>
<td>Recovery options and recovery jobs</td>
</tr>
<tr>
<td></td>
<td>scenario7.0.sh (UNIX)</td>
<td></td>
</tr>
<tr>
<td>Scenario 8</td>
<td>scenario8.0.bat (Windows)</td>
<td>Event-driven scheduling</td>
</tr>
<tr>
<td></td>
<td>scenario8.0.sh (UNIX)</td>
<td></td>
</tr>
<tr>
<td>Scenario 9</td>
<td>scenario9.0.bat (Windows)</td>
<td>Using variable tables</td>
</tr>
<tr>
<td></td>
<td>scenario9.0.sh (UNIX)</td>
<td></td>
</tr>
</tbody>
</table>

Creating and working with the production plan

After you have successfully populated the database, you are ready to run the Scenario 1, which creates the production plan. The production plan contains the database objects (jobs and job streams) that are eligible for scheduling.

Scenario 1 is a prerequisite to all other scenarios so you must run it first. The other scenarios can then be run in any order.

Most commands in the scenarios are given in their short form. Where this is the case, the full name of the command is shown in parentheses in each scenario description.
**Scenario 1: Creating the production plan and viewing its contents**

The scenario shows you how to:

- Create and extend a production plan
- Understand if a plan was created successfully
- View the contents of a plan

The scenario performs the following actions:

- Creates a production plan with a duration of 24 hours
- Inserts into the plan all the jobs and job streams that the tutorial already added in the database with their dependencies
- Views the contents of the plan

Commands used in the scenario in their run sequence:

1. `JnextPlan`
2. `conman sc` (showcpus)
3. `planman showinfo`
4. `conman ss @#SMPL@` (showschedules)

**Running the scheduling scenarios**

After creating the plan in Scenario 1, the other scenarios use the tutorial objects in the database by scheduling them in the plan. Each scenario explains different scheduling concepts. For each command used in the scenarios, the output is displayed on the screen.

**Note:** You can run the scenarios in any order because each scenario uses different objects. However, if you want to run the same scenario more than once in your sequence, you must reset the plan and run Scenario 1 again before you rerun the individual scenario. Perform these steps:

1. Run the following command:
   ```
   ResetPlan -scratch
   ```
2. Run the `scenario1.0.bat` in Windows or the `scenario1.0.sh` script in UNIX.

**Scenario 2: Starting and stopping Tivoli Workload Scheduler processes**

This scenario performs some basic administrative tasks. After each stop or start command, the status is displayed on the screen.

**Scenario tasks and concepts:**

- Stopping and starting the Tivoli Workload Scheduler engine
- Stopping and starting the event processor
- Stopping and starting the monitoring agent
- Viewing process status

Commands used in the scenario in their run sequence:

1. "conman stop"
2. "conman status"
3. "conman start"
4. "conman status"
5. "conman stopevtproc" (stopeventprocessor)
6. "conman startevtproc" (starteventprocessor)
7. "conman sc" (showcpus)
8. "conman stopmon;wait"
9. "conman startmon"
10. "conman sc" (showcpus)
For a detailed description of Tivoli Workload Scheduler processes and related commands, see the Tivoli Workload Scheduler: User’s Guide and Reference.

**Scenario 3: Scheduling basics, how jobs are scheduled, and run order of jobs**

This scenario performs basic scheduling tasks by showing how you schedule jobs and how you manage the scheduling sequence.

Scenario tasks and concepts:
- Running a job and a job stream on a workstation
- Viewing job status
- Viewing and changing the workstation limit
- Understanding the concept and purpose of dependent job streams and run order (FOLLOWS)
- Viewing dependency resolution during job runs

Commands used in the scenario in their run sequence:
1. "conman ss @#SMPL_SCHED_3@" (showschedules)
2. "composer disp sched=SMPL_SCHED_3_0_2"
3. "conman lc; 10;noask" (limit)
4. "conman sc" (showcpus)
5. "conman sj @#SMPL_SCHED_3_0_0.SMPL_JOB_3_0_0@" (showjobs)
6. "conman sj @#SMPL_SCHED_3_0_0.SMPL_JOB_3_0_0@" (showjobs)

**Scenario 4: Advanced scheduling, dependencies from prompts, files, and resources**

This scenario performs advanced scheduling tasks by showing different types of dependencies in action.

Scenario tasks and concepts:
- Viewing and managing prompt dependencies
- Viewing and managing resource dependencies
- Viewing and managing file dependencies
- Understanding resource contention between jobs

Commands used in the scenario in their run sequence:
1. "composer disp sched= @#SMPL_SCHED_4@"
2. "conman ss @#SMPL_SCHED_4@" (showschedules)
3. "conman sp @#SMLPRM4" (showprompts)
4. "conman reply SMLPRM4;y" (reply)
5. "conman sp @#SMLPRM4" (showprompts)
6. "conman sj @#SMPL_SCHED_4_0_0@" (showjobs)
7. "conman sj @#SMPL_SCHED_4_0_0@" (showjobs)
8. "conman sj @#SMPL_SCHED_4_0_0@" (showjobs)

**Scenario 5: Time dependencies and run cycles**

This scenario performs advanced scheduling using time dependencies and run cycles.

Scenario tasks and concepts:
- Managing time limits such as AT time and UNTIL time
- Releasing a time dependency
- Using run cycles to plan scheduling activities
Commands used in the scenario in their run sequence:

1. "conman sj @SMPL_SCHED_5_0_0.SMPL_JOB_5_0_0" (showjobs)
2. "conman ddj @SMPL_SCHED_5_0_1.SMPL_JOB_5_0_1;at;noask" (deldep)
3. "conman sj @SMPL_SCHED_5_0_1.SMPL_JOB_5_0_1" (showjobs)
4. "conman rj @SMPL_SCHED_5_0_1.SMPL_JOB_5_0_2" (release)
5. "conman sj @SMPL_SCHED_5_0_1.SMPL_JOB_5_0_2" (showjobs)
6. "conman ss @SMPL_SCHED_5-@" (showschedules)

**Scenario 6: Manual submission of jobs, job streams, and commands**

This scenario uses the submit command to insert jobs, job streams, and ad-hoc jobs in the plan.

Scenario tasks and concepts:
- Submitting a job in the current production plan
- Submitting a job stream in the current production plan
- Submitting a command in the current production plan
- Displaying the job, job stream, and command status in the plan

Commands used in the scenario in their run sequence:

1. "conman sbj @SMPL_JOB_SBJ;alias=SMPL_SBJ_ALIAS" (submit)
2. "conman sj @JOBS.SMPL_ALIAS" (showjobs)
3. "conman sbs @SMPL_SCHED_SBS;alias=SMPL_SBS_ALIAS" (submit)
4. "conman sj @SMPL_SBS_ALIAS" (showjobs)
5. "conman sbd "ver"; logon="SMPLUSER";alias=SMPL_SBD_ALIAS" (submit)
6. "conman sj @JOBS.SMPL_SBD_ALIAS" (showjobs)

**Note:** The value of the logon attribute in step 5 is specified by using a parameter object. For more information about parameters see the Tivoli Workload Scheduler: User's Guide and Reference.

**Scenario 7: Recovery options and recovery jobs**

This scenario shows some examples of recovery options and recovery jobs.

Scenario tasks and concepts:
- Defining and using the STOP, CONTINUE, and RERUN recovery options
- Understanding the use of recovery jobs to solve scheduling malfunctions

Commands used in the scenario in their run sequence:

1. "conman reply SMPLPRM7;y" (reply)
2. "conman sp SMPLPRM7" (showprompts)
3. "conman sj @SMPL_SCHED_7_0_1.0" (showjobs)
4. "conman sj @SMPL_SCHED_7_0_2.0" (showjobs)
5. "conman sj @SMPL_SCHED_7_0_3.0" (showjobs)

**Scenario 8: Event-driven scheduling**

This scenario shows some examples of event-driven scheduling.

Scenario tasks and concepts:
- Creating a rule and associating an action to the rule
Understanding the different rule types: Filter, Sequence, and Collection rules

Processing an action associated to a rule

Commands used in the scenario in their run sequence:
1. "composer disp erule=SMPL_FILTER_RULE" (display)
2. "planman deploy -scratch"
3. "conman sj @#JOBS.SMPL_SBJ_ALIAS2" (showjobs)

Scenario 9: Using variable tables

This scenario shows how you use variable tables to:

- Change the behavior of jobs and job streams based on why they are scheduled to run. For example, you can create a job that runs different commands for different operating systems.
- Change the behavior of jobs and job streams based on when they are scheduled to run, that is, on which days they run.

Commands used in the scenario in their run sequence:
1. "composer disp vartable=SMPL_VAR_TABLE_9_0_?" (display)
2. "composer disp vartable=MAIN_TABLE" (display)
3. "composer disp job=SMPL_JOB_9_1_1" (display)
4. "composer disp sched=SMPL_SCHED_9_1_1" (display)
5. "conman sj SMPL_SCHED_9_1_1(1000).SMPL_JOB_9_1_1;info" (showjobs)
6. "conman sj SMPL_SCHED_9_1_1(1200).SMPL_JOB_9_1_1;info" (showjobs)

Because the production plan has already been generated, you can see the following results:

- The job stream added for the run cycle associated to the SMPL_VAR_TABLE_9_0_2 variable table contains the SMPL_JOB_9_1_1 job that launches the default command.
- The job stream added for the run cycle associated to the SMPL_VAR_TABLE_9_0_1 variable table contains the SMPL_JOB_9_1_1 job that launches the command specified within the variable table.

Scenario 9 part 1: Using variable tables to run different commands using the same job definition:

This part shows how you use variable tables to create two job streams containing the same job definition to launch two different commands. The scenario performs the following steps:

- Creates two variable tables and defines variables inside them.
- Uses variables inside jobs.
- Defines two job streams
- Associates a different variable table to each job stream.

Commands used in the scenario in their run sequences:
1. "composer disp vartable=SMPL_VAR_TABLE_9_0_?" (display)
2. "composer disp job=SMPL_JOB_9_0_1" (display)
3. "composer disp sched=SMPL_SCHED_9_0_?" (display)
4. "conman sj SMPL_SCHED_9_0_1.SMPL_JOB_9_0_1;info" (showjobs)
5. "conman sj SMPL_SCHED_9_0_2.SMPL_JOB_9_0_1;info" (showjobs)
Because the production plan has already been generated, you can see the following results:

- The job added with the SMPL_SCHED_9_0_1 job stream contains the command to list the content of the TWSTutorial directory.
- The job added with the SMPL_SCHED_9_0_2 job stream contains the command to list the content of the TWS directory.

Scenario 9 part 2: Using variable tables to run different commands on different days:
This part shows how you use variable tables to have the same job stream containing two run cycles to launch two commands based on variable substitution. It create a job stream containing a job definition and two different run cycles that address two different variable tables. The scenario performs the following steps:

- Creates two variable tables and defines variables inside them.
- Uses variables inside jobs.
- Defines a job stream.
- Associates a different variable table to each run cycle.

Removing tutorial objects from the database
You can choose to keep the database objects in your environment to use them as templates for new objects. If, instead, you want to completely remove all tutorial objects from the database, perform the following steps:

1. Go to the TWS_home/TWS/TWSTutorial directory, where TWS_home is the home directory of the user for which you installed Tivoli Workload Scheduler.
2. Launch the tutorial utility installation script as follows:
   - In a Windows operating system:
     SAMPLEDBSETUP.CMD -uninstall
   - In a UNIX and Linux operating systems:
     sampledbsetup.sh -uninstall
Appendixes

Registry file

On UNIX operating systems, when you install Tivoli Workload Scheduler using the InstallShield wizard or the `twinst` script, a check is performed to determine whether there are other Tivoli Workload Scheduler instances already installed. The TWSRegistry.dat file stores the history of all instances installed. On Windows operating systems, this file is stored under the system drive directory, for example, `c:\winnt\system32`. On UNIX operating systems, this file is stored in the `/etc/TWS` path. The file contains the values of the following attributes that define a Tivoli Workload Scheduler installation:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProductID</td>
<td>TWS_ENGINE</td>
</tr>
<tr>
<td>PackageName</td>
<td>The name of the software package used to perform the installation.</td>
</tr>
<tr>
<td>InstallationPath</td>
<td>The absolute path of the Tivoli Workload Scheduler instance.</td>
</tr>
<tr>
<td>UserOwner</td>
<td>The owner of the installation.</td>
</tr>
<tr>
<td>MajorVersion</td>
<td>Tivoli Workload Scheduler version number.</td>
</tr>
<tr>
<td>MinorVersion</td>
<td>Tivoli Workload Scheduler release number.</td>
</tr>
<tr>
<td>MaintenanceVersion</td>
<td>Tivoli Workload Scheduler maintenance version number.</td>
</tr>
<tr>
<td>PatchVersion</td>
<td>The latest product patch number installed.</td>
</tr>
<tr>
<td>Agent</td>
<td>Any one of the following: standard agent, fault-tolerant agent, master domain manager.</td>
</tr>
<tr>
<td>FeatureList</td>
<td>The list of optional features installed.</td>
</tr>
<tr>
<td>LPName</td>
<td>The name of the software package block that installs the language pack.</td>
</tr>
<tr>
<td>LPLList</td>
<td>A list of all languages installed for the instance installed.</td>
</tr>
</tbody>
</table>

The following is an example of a TWSRegistry.dat file on a master domain manager:

```
/Tivoli/Workload_Scheduler/tws_nord_DN_objectClass=OU
/Tivoli/Workload_Scheduler/tws_nord_DN_PackageName=FP_Windows_tws_nord.8.3.00
/Tivoli/Workload_Scheduler/tws_nord_DN_MajorVersion=8
/Tivoli/Workload_Scheduler/tws_nord_DN_MinorVersion=2
/Tivoli/Workload_Scheduler/tws_nord_DN_PatchVersion=
/Tivoli/Workload_Scheduler/tws_nord_DN_FeatureList=TBSM
/Tivoli/Workload_Scheduler/tws_nord_DN_ProductID=TWS_ENGINE
/Tivoli/Workload_Scheduler/tws_nord_DN_ou=tws_nord
/Tivoli/Workload_Scheduler/tws_nord_DN_InstallationPath=c:\TWS\tws_nord
/Tivoli/Workload_Scheduler/tws_nord_DN_UserOwner=tws_nord
/Tivoli/Workload_Scheduler/tws_nord_DN_MaintenanceVersion=1
/Tivoli/Workload_Scheduler/tws_nord_DN_Agent=MDM
```
Using response files

All components of Tivoli Workload Scheduler and the Dynamic Workload Console that can be installed by the InstallShield wizard can also be installed silently, using a response file. A response file is a flat text list of property-value pairs each of which corresponding to a data item that the wizard needs to determine what is to be installed, where, and with what configuration. Silent installations can be used to install, upgrade or uninstall components locally, or remotely.

Tivoli Workload Scheduler and the Dynamic Workload Console components are provided with template response files, containing the appropriate properties to perform one installation, upgrade, or uninstallation action.

To perform a silent installation, provide the following command line arguments when running the wizard:
- options "<response_file_name> -silent"

The provided files are template files, so you are recommended to edit the properties appropriately, and then save a copy of the file with a file name which identifies the component to be installed and the system on which it is to be installed.

The properties have unique names and uses, and are described in the following sections. Many of them will contain default values that you can use. The defaults are not listed here as they may change, depending on which template file they are used in.

Note: Where the same template file is provided for Windows and UNIX platforms, default paths are supplied for both environments, with the keys duplicated and one commented out. Note that if you uncomment one and omit to comment the other, the wizard utilizes the last of the duplicated keys.

The Tivoli Workload Scheduler response file properties

This section describes the properties used in the Tivoli Workload Scheduler response files, in alphabetical order:

Note:
1. All values must be written between double quotation marks ("), for example: cpuCfgPanel.addFINAL="true"
2. Properties are written in mixed case for ease of reading, but are not case-sensitive
3. Keywords (for example, “true”) used in values, are not case-sensitive.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Permitted values</th>
</tr>
</thead>
<tbody>
<tr>
<td>checkJobsLoop.wait</td>
<td>Specify 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.</td>
<td>Integers or -1 for the product to wait indefinitely. The default is 60.</td>
</tr>
</tbody>
</table>
Table 24. Tivoli Workload Scheduler response file properties (continued)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Permitted values</th>
</tr>
</thead>
<tbody>
<tr>
<td>checkPrerequisites.</td>
<td>Stop the installation whenever any type of error or warning occurs during the prerequisite check. See “Checking prerequisites (UNIX and Linux)” on page 24 for more information on the prerequisite check.</td>
<td>true: Stop the installation whenever any type of error or warning occurs during the prerequisite check. False: Stop the installation whenever blocking errors occur during the prerequisite check. The default is false.</td>
</tr>
<tr>
<td>stopOnCheckPrereq</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cpuCfgPanel.addFINAL</td>
<td>Add the final job stream to the database. This option allows to perform automatic production plan extension at the end of each current production plan processing. By default, this box remains unchecked. This option is available only if you are installing a master domain manager.</td>
<td>true: Add the final job stream False: Do not add the final job stream</td>
</tr>
<tr>
<td>cpuCfgPanel.company</td>
<td>Company name.</td>
<td>See “Tivoli Workload Scheduler installation options” on page 61.</td>
</tr>
<tr>
<td>cpuCfgPanel.masterCPU</td>
<td>The name of the master domain manager workstation. When you are installing a master domain manager, this value must have the same value as cpuCfgPanel.thisCPU.</td>
<td>See “Tivoli Workload Scheduler installation options” on page 61.</td>
</tr>
<tr>
<td>cpuCfgPanel.jmPortNumberHttp</td>
<td>Specify if HTTPS is used for communication between the Tivoli Workload Scheduler agent and the Tivoli Workload Scheduler server, Tivoli Workload Scheduler for z/OS server, or the dynamic workload broker. Specify true for HTTPS and false for HTTP.</td>
<td>See “Installing an agent” on page 78.</td>
</tr>
<tr>
<td>cpuCfgPanel.tdwHostName</td>
<td>The fully qualified host name used by the Tivoli Workload Scheduler agent to connect to the dynamic workload broker.</td>
<td>See “Tivoli Workload Scheduler installation options” on page 61.</td>
</tr>
<tr>
<td>cpuCfgPanel.tcpPortNumber</td>
<td>The port used by netman on the system on which the component is installed.</td>
<td>See “Tivoli Workload Scheduler installation options” on page 61.</td>
</tr>
<tr>
<td>cpuCfgPanel.thisCPU</td>
<td>The name of the workstation where you are installing the component. When you are installing a master domain manager, this value must have the same value as cpuCfgPanel.masterCPU.</td>
<td>See “Tivoli Workload Scheduler installation options” on page 61.</td>
</tr>
<tr>
<td>cpuCfgPanel.jmPortNumber</td>
<td>The port used by the Workload Scheduler for z/OS server, or the dynamic workload broker to connect to the Tivoli Workload Scheduler dynamic agent.</td>
<td>See “Tivoli Workload Scheduler installation options” on page 61.</td>
</tr>
<tr>
<td>db2CheckPrereqs.db2Directory</td>
<td>The installation directory of the DB2 Enterprise Server or the DB2 Administration Client.</td>
<td>See “RDBMS installation options” on page 65.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Permitted values</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>db2ClientCfg.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>remoteNode</td>
<td>The remote node of the DB2 Administration Client.</td>
<td>See “RDBMS installation options” on page 65.</td>
</tr>
<tr>
<td>db2ClientCfg.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>remotePort</td>
<td>The TCP/IP port number that the remote DB2 server instance uses to communicate.</td>
<td>See “RDBMS installation options” on page 65.</td>
</tr>
<tr>
<td>db2ClientCfg.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>db2AdminUser</td>
<td>The user name of the administrator of the DB2 server instance.</td>
<td>See “RDBMS installation options” on page 65.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>db2ClientCfg.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>db2AdminPwd</td>
<td>The password of the DB2 server administrator user, or of the user with SYSADM or SYSCtrl authority.</td>
<td>See “RDBMS installation options” on page 65.</td>
</tr>
<tr>
<td>db2ClientCfg.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>db2LocalAdminUser</td>
<td>The user name of the DB2 administrator of the DB2 client instance.</td>
<td>See “RDBMS installation options” on page 65.</td>
</tr>
<tr>
<td>db2ClientCfg.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>db2LocalAdminPwd</td>
<td>The password of the DB2 administrator of the DB2 client instance.</td>
<td>See “RDBMS installation options” on page 65.</td>
</tr>
<tr>
<td>db2ClientCfg.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>twsDBUser</td>
<td>The user name of the DB2 user.</td>
<td>See “RDBMS installation options” on page 65.</td>
</tr>
<tr>
<td>db2ClientCfg.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>twsDBPwd</td>
<td>The password of the DB2 user.</td>
<td>See “RDBMS installation options” on page 65.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Permitted values</td>
</tr>
<tr>
<td>---------------------------</td>
<td>--------------------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td><code>db2ServerCfg. instanceName</code></td>
<td>The name of the DB2 server instance.</td>
<td>See “RDBMS installation options” on page 65</td>
</tr>
<tr>
<td><code>db2ServerCfg. instancePort</code></td>
<td>The TCP/IP port number used to communicate with the DB2 instance.</td>
<td>See “RDBMS installation options” on page 65</td>
</tr>
<tr>
<td><code>db2ServerCfg. db2AdminUser</code></td>
<td>The user name of the administrator of the DB2 Server.</td>
<td>See “RDBMS installation options” on page 65</td>
</tr>
<tr>
<td></td>
<td>If the DB2 administrator already created the database tables using the “Creating or upgrading the database tables if you are using DB2” on page 37 procedure, the user name is the one that the DB2 administrator specified in the <code>DB_USER</code> property in the <code>customizeDB2SQL.properties</code> file. The default value is: On Windows operating system <code>db2admin</code>. On UNIX and Linux operating systems <code>db2inst1</code>. If the DB2 administrator already upgraded the database tables using the “Creating or upgrading the database tables if you are using DB2” on page 37 procedure, the user name is the one that the DB2 administrator specified in the <code>DB_UPGRADE_USER</code> field. You must assign <code>SYSMON</code> authority to the user you specified in the <code>DB_UPGRADE_USER</code> field.</td>
<td></td>
</tr>
<tr>
<td><code>db2ServerCfg. db2AdminPwd</code></td>
<td>The password of the DB2 server administrator user, or of the user with <code>SYSADM</code> or <code>SYSCTRL</code> authority.</td>
<td>See “RDBMS installation options” on page 65</td>
</tr>
</tbody>
</table>
### Table 24. Tivoli Workload Scheduler response file properties (continued)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Permitted values</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>InstallationActions.</strong>&lt;br&gt;<strong>Install_Method</strong></td>
<td>Tivoli Workload Automation instance choice&lt;br&gt;Many of the Tivoli Workload Scheduler components, including the Dynamic Workload Console, must be installed in an instance of Tivoli Workload Automation (see “Instances of Tivoli Workload Automation” on page 27 for an explanation). This property lets you choose whether you want to install the component in a new instance (installing also the embedded WebSphere Application Server and other infrastructure support), or to install the component in an existing instance.&lt;br&gt;In the former case, the path you want to use for the new instance must be defined in the property <code>twsLocationPanel.directory</code>. In the latter case you must also identify the path of the existing instance, using the property:&lt;br&gt;<strong>InstallationActions.</strong>&lt;br&gt;<strong>TWA_INSTANCE_PATH</strong></td>
<td><strong>new</strong>&lt;br&gt;Install the Tivoli Workload Scheduler component in a new instance of Tivoli Workload Automation (and install the infrastructure support)&lt;br&gt;&lt;br&gt;<strong>ONTWA</strong>&lt;br&gt;Install the Tivoli Workload Scheduler component in an existing instance of Tivoli Workload Automation&lt;br&gt;Note: These values are case-sensitive.</td>
</tr>
<tr>
<td><strong>InstallationActions.</strong>&lt;br&gt;<strong>instanceID</strong></td>
<td>The command-line client is registered by Tivoli Workload Scheduler with an ID. When you upgrade an instance you must supply this ID.</td>
<td>It has the following format: <code>&lt;remoteHost&gt;;&lt;remoteUser&gt;</code></td>
</tr>
<tr>
<td><strong>InstallationActions.</strong>&lt;br&gt;<strong>TWA_INSTANCE_PATH</strong></td>
<td>Tivoli Workload Automation instance path.&lt;br&gt;Identifies the path where an instance of Tivoli Workload Automation has already been installed.</td>
<td>See “Tivoli Workload Scheduler installation options” on page 61.</td>
</tr>
<tr>
<td><strong>InstallationActions.</strong>&lt;br&gt;<strong>twsUser</strong></td>
<td>The existing TWSUser.&lt;br&gt;Identifies the TWSUser of an existing component you are upgrading.</td>
<td></td>
</tr>
<tr>
<td><strong>installationAgent</strong>&lt;br&gt;<strong>Components.</strong>&lt;br&gt;<strong>addEclipse</strong></td>
<td>Adds the Java runtime to run job types with advanced options to the dynamic agent. The runtime environment is used to:&lt;br&gt;• 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.&lt;br&gt;• Enable the capability to remotely run, from the agent, the dynamic workload broker resource command on the server.</td>
<td><strong>true</strong>&lt;br&gt;Adds the Java runtime to run job types with advanced options.&lt;br&gt;&lt;br&gt;<strong>false</strong>&lt;br&gt;To not the Java runtime to run job types with advanced options</td>
</tr>
<tr>
<td><strong>installationAgent</strong>&lt;br&gt;<strong>Components.</strong>&lt;br&gt;<strong>addTdwb</strong></td>
<td>Adds to the Tivoli Workload Scheduler agent the capability to run dynamic workload.</td>
<td><strong>true</strong>&lt;br&gt;To add the capability to run dynamic workload.&lt;br&gt;&lt;br&gt;<strong>false</strong>&lt;br&gt;To not add the capability to run dynamic workload.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Permitted values</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>installationAgent</td>
<td>Components.</td>
<td></td>
</tr>
<tr>
<td>instanceType</td>
<td>The type of agent to install.</td>
<td>LWA: Agent to run workload from a z/OS environment in a distributed environment.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FTA: Fault-tolerant and domain manager agents</td>
</tr>
<tr>
<td>installationComponents.</td>
<td>Tivoli Workload Automation instance type.</td>
<td>MDM: master domain manager</td>
</tr>
<tr>
<td>instanceType</td>
<td>The type of component to install.</td>
<td>BKM: backup master domain manager</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DDM: dynamic domain manager</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BDM: backup dynamic domain manager</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AGENT: Fault-tolerant agent, dynamic agent, or domain manager</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CLI: command-line client</td>
</tr>
<tr>
<td>InstallLocation</td>
<td>Installation path for the Integration Workbench.</td>
<td>Any fully qualified path outside an instance of Tivoli Workload Automation.</td>
</tr>
<tr>
<td>IsOnlyFTAConnector</td>
<td>To uninstall.</td>
<td>Yes: Uninstall just the distributed connector.</td>
</tr>
<tr>
<td>ToUninstall.</td>
<td></td>
<td>No: Uninstall the fault-tolerant agent, the dynamic agent, and the distributed connector.</td>
</tr>
<tr>
<td>licenseAccepted</td>
<td>Accept license agreement.</td>
<td>true: To accept the license agreement.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>false: To not accept the license agreement. In this event the component is not installed.</td>
</tr>
<tr>
<td>TdwbConfig.</td>
<td></td>
<td>See “Tivoli Workload Scheduler installation options” on page 61.</td>
</tr>
<tr>
<td>tdwCpuName</td>
<td>The dynamic workload broker workstation that you will create in the Tivoli Workload Scheduler database</td>
<td></td>
</tr>
<tr>
<td>TdwbConfig.</td>
<td></td>
<td>See “Tivoli Workload Scheduler installation options” on page 61.</td>
</tr>
<tr>
<td>tdwCpuPort</td>
<td>The port of the dynamic workload broker workstation that you will create in the Tivoli Workload Scheduler database. The Tivoli Workload Scheduler engine and the dynamic workload broker component communicate using this port.</td>
<td></td>
</tr>
<tr>
<td>oracleCheckPrereqs.</td>
<td></td>
<td>See “Installing for an Oracle database” on page 69.</td>
</tr>
<tr>
<td>oracleDirectory</td>
<td>The installation directory of the Oracle database.</td>
<td></td>
</tr>
<tr>
<td>CommunicationInfo.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>netServiceName</td>
<td>The name used by clients to identify an Oracle Net server and the specific system identifier or database for the Oracle Net connection.</td>
<td></td>
</tr>
</tbody>
</table>
Table 24. Tivoli Workload Scheduler response file properties (continued)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Permitted values</th>
</tr>
</thead>
<tbody>
<tr>
<td>oracleServer</td>
<td>The database administrator user name (such as SYSTEM) required to authenticate to the Oracle database.</td>
<td>See “Installing for an Oracle database” on page 69.</td>
</tr>
<tr>
<td>CommunicationInfo.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>oracleAdminUser</td>
<td>If the ORACLE administrator already created the database tables using the “Creating or upgrading the database tables if you are using Oracle” on page 47, the user name is the one that the ORACLE administrator specified in the MDL_USER property of the customizeORACLESQ.properties file.</td>
<td></td>
</tr>
<tr>
<td>oracleServer</td>
<td>The database administrator user password required to authenticate to the Oracle database.</td>
<td>See “Installing for an Oracle database” on page 69.</td>
</tr>
<tr>
<td>CommunicationInfo.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>oracleAdminPwd</td>
<td></td>
<td></td>
</tr>
<tr>
<td>recovInstReg.run</td>
<td>Specify to upgrade workstations with corrupted registry files without having to reinstall the product. Use this option also to upgrade a cluster environment from any node of the cluster, not only from the one where you installed Tivoli Workload Scheduler. This option is particularly useful when the cluster node where the product is installed is unavailable or in an inconsistent state. If you specify this option, Tivoli Workload Scheduler re-creates the installation registries and the Software Distribution information.</td>
<td>true To upgrade workstations with corrupted registry files without having to reinstall the product. The default is false.</td>
</tr>
<tr>
<td>SDK_ECLIPSE_BUNDLED</td>
<td>Integration Workbench is to be installed with the bundled version of Eclipse.</td>
<td>true To install Integration Workbench with the bundled version of Eclipse</td>
</tr>
<tr>
<td></td>
<td>Whichever setting you choose, the property SDK_UPDATESITE must have the opposite setting.</td>
<td>false To not install Integration Workbench with the bundled version of Eclipse.</td>
</tr>
<tr>
<td>SDK_UPDATESITE</td>
<td>Integration Workbench is to be installed as an Eclipse site, without the bundled version of Eclipse.</td>
<td>true To install Integration Workbench as an Eclipse site.</td>
</tr>
<tr>
<td></td>
<td>Whichever setting you choose, the property SDK_ECLIPSE_BUNDLED must have the opposite setting.</td>
<td>false To not install Integration Workbench as an Eclipse site.</td>
</tr>
<tr>
<td>selectRDBMSPanel.</td>
<td>Choose which type of RDBMS support you want to use, DB2 or Oracle.</td>
<td>“DB2” or “Oracle”, not case-sensitive.</td>
</tr>
<tr>
<td>rdbmsSelected</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TdwbConfig.</td>
<td>The dynamic workload broker workstation that you will create in the Tivoli Workload Scheduler database</td>
<td>See “Tivoli Workload Scheduler installation options” on page 61.</td>
</tr>
<tr>
<td>tdwbcpuName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Permitted values</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TdwbConfig. tdwbCpuPort</td>
<td>The port of the dynamic workload broker workstation that you will create in the Tivoli Workload Scheduler database. The Tivoli Workload Scheduler engine and the dynamic workload broker component communicate using this port.</td>
<td>See “Tivoli Workload Scheduler installation options” on page 61.</td>
</tr>
<tr>
<td>twsClcCfgPanel. password</td>
<td>Password of the user identified in twsClcCfgPanel.user.</td>
<td></td>
</tr>
<tr>
<td>twsClcCfgPanel. remoteHost</td>
<td>The host name or IP address of the workstation where the master domain manager is installed.</td>
<td></td>
</tr>
<tr>
<td>twsClcCfgPanel. remotePort</td>
<td>The listening port of the workstation where the master domain manager is installed.</td>
<td></td>
</tr>
<tr>
<td>twsClcCfgPanel. user</td>
<td>The user ID used to access the master domain manager. Usually the &lt;TWS_user&gt;.</td>
<td></td>
</tr>
<tr>
<td>TWSCLILanguagesPanel. all</td>
<td>Choose command-line client languages: all languages. When installing the language pack on the command-line client, you can select to install all languages, using this property, or select the specific languages you require (see following property)</td>
<td>true All languages are installed. false You want to install specific languages using the TWSCLILanguagesPanel. &lt;language&gt; properties.</td>
</tr>
<tr>
<td>TWSCLILanguagesPanel. chineseSimplified chineseTraditional german french italian japanese korean portuguese spanish</td>
<td>Choose command-line client languages: specific languages. When installing the language pack on the command-line client, you can select to install all languages, using the TWSCLILanguagesPanel.all property, or select the specific languages you require, using one or more of these properties.</td>
<td>For each property: true The selected language is installed. false The selected language is not installed.</td>
</tr>
<tr>
<td>twsCLILocationPanel. directory</td>
<td>The path where you want to install the command-line client.</td>
<td>Any valid, fully qualified path outside any instance of Tivoli Workload Automation.</td>
</tr>
<tr>
<td>twsDBCfg.dbName</td>
<td>The name of the DB2 database.</td>
<td>See “RDBMS installation options” on page 65.</td>
</tr>
<tr>
<td>twsDBCfg.tablespaceName</td>
<td>The name of the DB2 instance tablespace.</td>
<td>See “RDBMS installation options” on page 65.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Permitted values</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>twsDBCfg.tablespacePath</td>
<td>The relative path of the DB2 table space.</td>
<td>See “RDBMS installation options” on page 65.</td>
</tr>
<tr>
<td>twsDBCfg.reportTablespaceName</td>
<td>The name of the table space for storing report data.</td>
<td>See “RDBMS installation options” on page 65.</td>
</tr>
<tr>
<td>twsDBCfg.reportTablespacePath</td>
<td>The path of the table space for storing report data.</td>
<td>See “RDBMS installation options” on page 65.</td>
</tr>
<tr>
<td>twsLocationPanel.directory</td>
<td>The path where you want to install the fresh Tivoli Workload Scheduler component.</td>
<td>Any valid, fully qualified Tivoli Workload Automation instance path.</td>
</tr>
<tr>
<td>twsLocationPanel.symLinkOption</td>
<td>Choose whether to create symbolic links (see Table 2 on page 26 for more details).</td>
<td>true Symbolic links are created. false Symbolic links are not created.</td>
</tr>
<tr>
<td>twsOracleDbCfg.twsDBUser</td>
<td>The owner of the Tivoli Workload Scheduler schema.</td>
<td>See “Installing for an Oracle Database” on page 69.</td>
</tr>
<tr>
<td>twsOracleDbCfg.twsDBPwd</td>
<td>The database administrator user password required to authenticate to the Oracle database.</td>
<td>See “Installing for an Oracle Database” on page 69.</td>
</tr>
<tr>
<td>twsOracleDbCfg.twsDataTablespace</td>
<td>The name that identifies the Tivoli Workload Scheduler data table space.</td>
<td>See “Installing for an Oracle Database” on page 69.</td>
</tr>
<tr>
<td>twsOracleDbCfg.twsReportTablespace</td>
<td>The name that identifies the Tivoli Workload Scheduler table space where report data is to be stored.</td>
<td>See “Installing for an Oracle Database” on page 69.</td>
</tr>
<tr>
<td>twsOracleDbCfg.twsTempTablespace</td>
<td>The name that identifies the Tivoli Workload Scheduler temporary table space.</td>
<td>See “Installing for an Oracle Database” on page 69.</td>
</tr>
<tr>
<td>twsOracleDbCfg.isPartitioned</td>
<td>Specify whether the event-driven workload automation database schema is to be created using the Oracle Partitioning feature.</td>
<td>true The Oracle Partitioning feature is used when creating the event-driven workload automation database schema. false The Oracle Partitioning feature is NOT used when creating the event-driven workload automation database schema.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Permitted values</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>twsPortsPanel.portAdmin</td>
<td>Administration HTTP transport port.</td>
<td>See &quot;WebSphere Application Server installation options&quot; on page 64 for more details.</td>
</tr>
<tr>
<td>twsPortsPanel.portAdminSec</td>
<td>Administration HTTPS transport port.</td>
<td>See &quot;WebSphere Application Server installation options&quot; on page 64 for more details.</td>
</tr>
<tr>
<td>twsPortsPanel.portEIflf</td>
<td>Event processor port</td>
<td>See &quot;WebSphere Application Server installation options&quot; on page 64 for more details.</td>
</tr>
<tr>
<td>twsPortsPanel.portHTTP</td>
<td>HTTP transport port</td>
<td>See &quot;WebSphere Application Server installation options&quot; on page 64 for more details.</td>
</tr>
<tr>
<td>twsPortsPanel.portHTTPS</td>
<td>HTTPS transport port</td>
<td>See &quot;WebSphere Application Server installation options&quot; on page 64 for more details.</td>
</tr>
<tr>
<td>twsPortsPanel.portMtlAuth</td>
<td>CSIv2 Client Authentication Listener port</td>
<td>See &quot;WebSphere Application Server installation options&quot; on page 64 for more details.</td>
</tr>
<tr>
<td>twsPortsPanel.portORB</td>
<td>ORB Listener port</td>
<td>See &quot;WebSphere Application Server installation options&quot; on page 64 for more details.</td>
</tr>
<tr>
<td>twsPortsPanel.portRMI</td>
<td>Bootstrap port</td>
<td>See &quot;WebSphere Application Server installation options&quot; on page 64 for more details.</td>
</tr>
<tr>
<td>twsPortsPanel.portSAS</td>
<td>SAS Server Authentication Listener port</td>
<td>See &quot;WebSphere Application Server installation options&quot; on page 64 for more details.</td>
</tr>
<tr>
<td>twsPortsPanel.portSOAP</td>
<td>SOAP connector port</td>
<td>See &quot;WebSphere Application Server installation options&quot; on page 64 for more details.</td>
</tr>
<tr>
<td>twsPortsPanel.portSrvAuth</td>
<td>CSIv2 Server Authentication Listener port</td>
<td>See &quot;WebSphere Application Server installation options&quot; on page 64 for more details.</td>
</tr>
</tbody>
</table>
| twsUpgradePanel.backupOldInstance | Determines if the existing instance of a component is to be backed up during an upgrade. | true The existing instance is backed up.    
<p>|                           |                                                                             | false The existing instance is not backed up.                                    |</p>
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Permitted values</th>
</tr>
</thead>
<tbody>
<tr>
<td>twsUpgradePanel. bckpDirectory</td>
<td>The backup directory used if the existing instance of a component is to be backed up during an upgrade.</td>
<td>Any valid fully qualified path outside the path of any existing Tivoli Workload Scheduler component.</td>
</tr>
<tr>
<td>twsUpgradePanel. bckpProfileDirectory</td>
<td>The backup directory for the application server profile (used when upgrading the master domain manager or backup master domain manager from version 8.3 or 8.4).</td>
<td>Any valid fully qualified path outside the path of any existing Tivoli Workload Scheduler component.</td>
</tr>
<tr>
<td>twsUpgradePanel. dumpDirectory</td>
<td>The migration directory used when upgrading a master domain manager from version 8.2.x. The database is exported to this directory as flat text files, and then imported into the new RDBMS support.</td>
<td>Any valid fully qualified path outside the path of any existing Tivoli Workload Scheduler component.</td>
</tr>
<tr>
<td>userUnixCfgPanel. inputUserName</td>
<td>The user ID of the &lt;TWS_user&gt; (on UNIX).</td>
<td>The ID must already exist on the system where the silent wizard will be run.</td>
</tr>
<tr>
<td>userUnixCfgPanel. twsPassword</td>
<td>The password of the &lt;TWS_user&gt; (on UNIX).</td>
<td></td>
</tr>
<tr>
<td>userUnixCfgPanel. wasPassword</td>
<td>If the Dynamic Workload Console has already been installed on an existing instance of Tivoli Workload Automation, supply the password of the WebSphere Application Server user ID of the embedded WebSphere Application Server that you configured when you installed the Dynamic Workload Console (on UNIX).</td>
<td>This will normally be the password of the &lt;TWS_user&gt;, unless you have changed it in the embedded WebSphere Application Server.</td>
</tr>
<tr>
<td>userUnixCfgPanel. wasUserName</td>
<td>If the Dynamic Workload Console has already been installed on an existing instance of Tivoli Workload Automation, supply the WebSphere Application Server user ID of the embedded WebSphere Application Server that you configured when you installed the Dynamic Workload Console (on UNIX).</td>
<td>This will normally be the &lt;TWS_user&gt;, unless you have changed it in the embedded WebSphere Application Server.</td>
</tr>
<tr>
<td>userWinCfgPanel. inputUserName</td>
<td>The ID of the &lt;TWS_user&gt; - the user that will &quot;own&quot; the agent on the agent workstation (on Windows).</td>
<td>If this user does not already exist, it will be created. In this case, the format of the ID must follow the rules for User IDs on the computer where it is to be created.</td>
</tr>
<tr>
<td>userWinCfgPanel. twsPassword</td>
<td>The password of the &lt;TWS_user&gt; (on Windows).</td>
<td>If the user is to be created, the format of the password must follow the rules for passwords on the computer where it is to be created.</td>
</tr>
<tr>
<td>userWinCfgPanel. wasPassword</td>
<td>If the Dynamic Workload Console has already been installed on an existing instance of Tivoli Workload Automation, supply the password of the WebSphere Application Server user ID of the embedded WebSphere Application Server that you configured when you installed the Dynamic Workload Console (on Windows).</td>
<td>This will normally be the password of the &lt;TWS_user&gt;, unless you have changed it in the embedded WebSphere Application Server.</td>
</tr>
</tbody>
</table>
Table 24. Tivoli Workload Scheduler response file properties (continued)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Permitted values</th>
</tr>
</thead>
<tbody>
<tr>
<td>userWinCfgPanel.</td>
<td>If the Dynamic Workload Console has already been installed on an existing</td>
<td>This will normally be the &lt;TWS_user&gt;, unless you have changed it in the embedded</td>
</tr>
<tr>
<td>wasUserName</td>
<td>instance of Tivoli Workload Automation, supply the WebSphere Application</td>
<td>WebSphere Application Server.</td>
</tr>
<tr>
<td></td>
<td>Server user ID of the embedded WebSphere Application Server that you</td>
<td></td>
</tr>
<tr>
<td></td>
<td>configured when you installed the Dynamic Workload Console (on Windows).</td>
<td></td>
</tr>
</tbody>
</table>

The Dynamic Workload Console response file properties

This section describes the properties used in the Dynamic Workload Console response files, in alphabetical order:

**Note:**

1. All values must be written between double quotation marks ("), for example: InstallationActions.INSTALL_METHOD="new"
2. Property names are written in mixed case for ease of reading, but are not case-sensitive
3. Keywords used in values are not case-sensitive.

Table 25. Dynamic Workload Console response file properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Permitted values</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOOTSTRAP_ADDRESS</td>
<td>The bootstrap port.</td>
<td>See “Advanced installation” on page 221 for more details.</td>
</tr>
<tr>
<td>CREATE_WAS_SERVICE</td>
<td>On Windows, the embedded WebSphere Application Server can be defined to</td>
<td>true A Windows service is created to automatically start the embedded WebSphere</td>
</tr>
<tr>
<td></td>
<td>start automatically at system startup. To do this, set this property,</td>
<td>Application Server</td>
</tr>
<tr>
<td></td>
<td>which creates a Windows service that starts up the embedded WebSphere</td>
<td>false The Windows service is not created</td>
</tr>
<tr>
<td></td>
<td>Application Server.</td>
<td></td>
</tr>
<tr>
<td>CSIV2_SSL_MUTUALAUTH_LISTEN</td>
<td>CSiv2 Client Authentication Listener port.</td>
<td>See “Advanced installation” on page 221 for more details.</td>
</tr>
<tr>
<td>_ER_ADDRESS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSIV2_SSL_SERVERAUTH_LISTEN</td>
<td>CSiv2 Server Authentication Listener port</td>
<td>See “Advanced installation” on page 221 for more details.</td>
</tr>
<tr>
<td>ER_ADDRESS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCS_UNICAST_ADDRESS</td>
<td>The DCS Unicast port.</td>
<td>See “Advanced installation” on page 221 for more details.</td>
</tr>
</tbody>
</table>
### Table 25. Dynamic Workload Console response file properties (continued)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Permitted values</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ENABLE_TDWB</strong></td>
<td>Enable Dynamic Workload Broker</td>
<td>true: Gives the administrator access to Dynamic Workload Broker</td>
</tr>
<tr>
<td></td>
<td>The Dynamic Workload Console can be used to access either of the following:</td>
<td>false: Denies the administrator access to Dynamic Workload Broker</td>
</tr>
<tr>
<td></td>
<td>• Tivoli Workload Scheduler (includes Tivoli Workload Scheduler for z/OS)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Dynamic workload broker</td>
<td></td>
</tr>
<tr>
<td></td>
<td>All users must be given specific access to one or both of these products.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>It is useful to give these access rights to the WebSphere Application Server administrator from the outset, so that the administrator can immediately perform any tasks that might be required:</td>
<td></td>
</tr>
<tr>
<td><strong>ENABLE_TWS</strong></td>
<td>Enable Tivoli Workload Scheduler</td>
<td>true: Gives the administrator access to Tivoli Workload Scheduler</td>
</tr>
<tr>
<td></td>
<td>See the description of &quot;ENABLE_TDWB&quot;</td>
<td>false: Denies the administrator access to Tivoli Workload Scheduler</td>
</tr>
<tr>
<td><strong>INSTALL_METHOD</strong></td>
<td>Installation instance choice</td>
<td>true: Install the Dynamic Workload Console in a new instance of Tivoli Workload Automation (and install the infrastructure support). Use this value also when upgrading an existing instance of the Dynamic Workload Console.</td>
</tr>
<tr>
<td></td>
<td>The Dynamic Workload Console must be installed in an instance of Tivoli Workload Automation (see <a href="#">Instances of Tivoli Workload Automation</a> for an explanation). This property lets you choose whether you want to install the component in a new instance (installing also the embedded WebSphere Application Server and other infrastructure support), or an existing instance.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>In the former case, the path you want to use for the new instance must be defined in the property IS_DESTINATION. In the latter case you must also identify the path of the existing instance, using the property: TWA_INSTANCE_PATH.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This property also lets you install the Dynamic Workload Console outside the Tivoli Workload Automation structure, on your own external supported version of WebSphere Application Server. In this case, the path must be supplied using the property ISC_APPSERVER_DIR.</td>
<td></td>
</tr>
<tr>
<td><strong>IPC_CONNECTOR_ADDRESS</strong></td>
<td>The IPC connector.</td>
<td>See &quot;Advanced installation&quot; on page 221 for more details.</td>
</tr>
</tbody>
</table>

---

IBM Tivoli Workload Scheduler: Planning and Installation
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Permitted values</th>
</tr>
</thead>
</table>
| IS_BACKUP_DIR      | Backup directory for upgrade  
When upgrading the Dynamic Workload Console, the wizard needs to back up the application server configuration while it is upgrading embedded WebSphere Application Server (part of the Dynamic Workload Console upgrade process). | Any valid, fully qualified path outside: any existing instance of Tivoli Workload Automation, and the installation path of the Embedded Version of WebSphere Application Server                                                                                           |
| IS_DESTINATION     | Console installation path  
**On a new instance of Tivoli Workload Automation:** the path of a new instance of Tivoli Workload Automation where the Dynamic Workload Console is to be installed.  
**On your existing external instance of WebSphere Application Server:** when installing the Dynamic Workload Console on your own external version of WebSphere Application Server, supply the console installation path.  
This does not have to be a path related to the instance of the WebSphere Application Server on which you are going to install it. The path must not be within an instance of Tivoli Workload Automation. | Any valid, fully qualified path outside any existing instance of Tivoli Workload Automation.                                                                                                                                  |
| IS_UPGRADE         | Boolean property that determines whether the wizard is being run to upgrade an existing instance.                                                                                                             | **true**  
The wizard will use the supplied properties to upgrade an existing instance of the Dynamic Workload Console  
**false**  
The wizard will use the supplied properties to install an instance of the Dynamic Workload Console                                                                                                               |
| ISC_ADMIN_FULL_USER| Your WebSphere Application Server administrator user ID  
**On a new instance of Tivoli Workload Automation:** supply the user ID to be used for the Integrated Solutions Console administration user  
**On your existing external instance of WebSphere Application Server:** when installing, upgrading, or uninstalling the Dynamic Workload Console on your own external version of WebSphere Application Server, supply the existing user ID of the Integrated Solutions Console administration user. | The user ID must exist.                                                                                                                                                                                       |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Permitted values</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISC_ADMIN_PASSWORD</td>
<td>Your WebSphere Application Server administrator user password</td>
<td></td>
</tr>
</tbody>
</table>
### Table 25. Dynamic Workload Console response file properties (continued)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Permitted values</th>
</tr>
</thead>
<tbody>
<tr>
<td>WAS_CELL_NAME</td>
<td>The WebSphere Application Server cell name</td>
<td>See “Installing on your existing instance of Tivoli Integrated Portal” on page 223 for more details.</td>
</tr>
<tr>
<td></td>
<td>The external WebSphere Application Server cell name.</td>
<td></td>
</tr>
<tr>
<td>WAS_NODE_NAME</td>
<td>The WebSphere Application Server node name</td>
<td>See “Installing on your existing instance of Tivoli Integrated Portal” on page 223 for more details.</td>
</tr>
<tr>
<td></td>
<td>The external WebSphere Application Server node name.</td>
<td></td>
</tr>
<tr>
<td>WAS_PROFILE_NAME</td>
<td>The WebSphere Application Server profile name</td>
<td>See “Installing on your existing instance of Tivoli Integrated Portal” on page 223 for more details.</td>
</tr>
<tr>
<td></td>
<td>The external WebSphere Application Server profile name.</td>
<td></td>
</tr>
<tr>
<td>WAS_SERVER_NAME</td>
<td>The WebSphere Application Server server name</td>
<td>See “Installing on your existing instance of Tivoli Integrated Portal” on page 223 for more details.</td>
</tr>
<tr>
<td></td>
<td>The external WebSphere Application Server server name.</td>
<td></td>
</tr>
<tr>
<td>WC_adminhost</td>
<td>Administrative console</td>
<td>See “Advanced installation” on page 221 for more details.</td>
</tr>
<tr>
<td>WC_adminhost_secure</td>
<td>Administrative Console Secure</td>
<td>See “Advanced installation” on page 221 for more details.</td>
</tr>
<tr>
<td>WC_defaulthost</td>
<td>HTTP transport</td>
<td>See “Advanced installation” on page 221 for more details.</td>
</tr>
<tr>
<td>WC_defaulthost_secure</td>
<td>HTTPS transport</td>
<td>See “Advanced installation” on page 221 for more details.</td>
</tr>
</tbody>
</table>

### The Job Brokering Definition Console response file properties

This section describes the properties used in the Job Brokering Definition Console response files:

**Note:**

1. All values must be written between double quotation marks (".
2. Property names are written in mixed case for ease of reading, but are not case-sensitive
3. Keywords used in values are not case-sensitive.
### Table 26. Job Brokering Definition Console response file properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Permitted values</th>
</tr>
</thead>
<tbody>
<tr>
<td>licenseAccepted</td>
<td>Accept license agreement. To install the Job Brokering Definition Console using a response file, you must explicitly accept the license agreement, a copy of which is in the License directory of the product install media (DVD or downloaded image). The license must be accepted before installation. This value must equal <strong>true</strong> for the installation to be successful.</td>
<td><strong>true</strong> To accept the license agreement. <strong>false</strong> To not accept the license agreement. In this event, the Job Brokering Definition Console is not installed.</td>
</tr>
<tr>
<td>installLocation</td>
<td>Installation path for the Job Brokering Definition Console.</td>
<td>Any fully qualified path.</td>
</tr>
</tbody>
</table>

---

### Installing and upgrading Tivoli Workload Scheduler Integration Workbench

Use Tivoli Workload Scheduler Integration Workbench to develop event and action plug-ins that extend the capabilities of Tivoli Workload Scheduler event-driven workload automation. You can also create Java applications that use the Tivoli Workload Scheduler Java API.

You can install Tivoli Workload Scheduler Integration Workbench with the bundled version of Eclipse or with an existing instance of Eclipse. The minimum supported version is Eclipse HELIOS 3.6 with the Plug-in Development Environment (PDE) installed.

This appendix contains the following sections:

- “Installing Tivoli Workload Scheduler Integration Workbench with the bundled version of Eclipse” on page 308
- “Installing Tivoli Workload Scheduler Integration Workbench with an existing instance of Eclipse using the Eclipse Site” on page 309
- “Installing Tivoli Workload Scheduler Integration Workbench with an existing instance of Eclipse using the remote Eclipse Site” on page 309
- “Upgrading Tivoli Workload Scheduler Integration Workbench installed with the bundled version of Eclipse” on page 309
- “Upgrading Tivoli Workload Scheduler Integration Workbench installed as a plug-in” on page 310

**Note:** If you are working with an exported display, to access the Tivoli Workload Scheduler Integration Workbench readme file you must have a browser open and running.

---

### Installing Tivoli Workload Scheduler Integration Workbench with the bundled version of Eclipse

If you do not have the required version of Eclipse on your computer, you can install Tivoli Workload Scheduler Integration Workbench bundled with Eclipse for local use.
To install Tivoli Workload Scheduler Integration Workbench with the bundled version of Eclipse, perform the following actions:

1. Either, from the installation DVD, navigate to the IntegrationWorkbench directory and run the setup file appropriate to your operating system.
   Or, start the launchpad as described in “Launchpad” on page 30 and select the Tivoli Workload Scheduler Integration Workbench installation.

2. Follow the installation wizard and when prompted, select Install Integration Workbench.

Installing Tivoli Workload Scheduler Integration Workbench with an existing instance of Eclipse using the Eclipse Site

If you have the required version of Eclipse, you can install Tivoli Workload Scheduler Integration Workbench as a plug-in on the existing instance. Users across the network can access Tivoli Workload Scheduler Integration Workbench as an Eclipse site.

To install the current version of Tivoli Workload Scheduler Integration Workbench into your existing instance of Eclipse, perform the following actions:

1. Either, from the installation DVD, navigate to the IntegrationWorkbench directory and run the setup file appropriate to your operating system.
   Or, start the launchpad as described in “Launchpad” on page 30 and select the Tivoli Workload Scheduler Integration Workbench installation.

2. Follow the installation wizard and when prompted, select Install Eclipse site.

Note: For information about using the plug-in, see the readme document in Eclipse.

Installing Tivoli Workload Scheduler Integration Workbench with an existing instance of Eclipse using the remote Eclipse Site

If you already have Eclipse on your computer, you can install Tivoli Workload Scheduler Integration Workbench as a plug-in for an existing instance of Eclipse, using the IBM remote Eclipse Site for Tivoli Workload Scheduler Integration Workbench.

To install the current version of Tivoli Workload Scheduler Integration Workbench into your existing instance of Eclipse, use the Eclipse Software Update feature specifying in the Site field, the following link:

ftp://public.dhe.ibm.com/software/tivoli_support/misc/TWS/SDK/

Note: For information about using the plug-in, see the readme document in Eclipse.

Upgrading Tivoli Workload Scheduler Integration Workbench installed with the bundled version of Eclipse

If you installed Tivoli Workload Scheduler Integration Workbench version 8.5.0 or 8.5.1 using the bundle version of Eclipse, you can upgrade it using the Eclipse Software Updates feature. To upgrade Tivoli Workload Scheduler Integration Workbench version 8.5.0 or 8.5.1, perform the following actions:

1. Close Tivoli Workload Scheduler Workbench version 8.5.0 or 8.5.1.
2. Move to `TivoliWorkloadSchedulerIntegrationWorkbenchversion_installation_directory/eclipse/plugins`.

3. In this directory, remove the `org.eclipse.ecf.identity.jar_<version>` file.

4. In the `TivoliWorkloadSchedulerIntegrationWorkbenchversion_installation_directory/eclipse/features/com.ibm.tws.sdk_<version>/feature.xml` file, change the URL already present in the `update` tag, with the following URL:

   ```xml
   <url>
     update label="%updatesite"
     url="ftp://public.dhe.ibm.com/software/tivoli_support/misc/TWS/SDK/
   </url>
   ```

5. Save and close the file.


7. Upgrade your old version of Tivoli Workload Scheduler Integration Workbench using the Eclipse Software Updates feature.

   **Note:** For information about using the plug-in, see the readme document in Eclipse.

### Upgrading Tivoli Workload Scheduler Integration Workbench installed as a plug-in

If you installed Tivoli Workload Scheduler Integration Workbench version 8.5.0 or 8.5.1 on an existing version of Eclipse as a plug-in, you can upgrade it using the Eclipse Software Updates feature, by performing the following actions:

1. Close the Eclipse where you installed Tivoli Workload Scheduler Integration Workbench version 8.5.0 or 8.5.1.

2. Move to the `eclipse_installation_directory/eclipse/features` directory.

3. Update the `com.ibm.tws.sdk_<version>/feature.xml` with the following address:

   ```xml
   ftp://public.dhe.ibm.com/software/tivoli_support/misc/TWS/SDK/
   ```

4. Upgrade the old version of Tivoli Workload Scheduler Integration Workbench using the Eclipse Software Updates feature.

   **Note:** For information about using the plug-in, see the readme document in Eclipse.

### Discovering installed products

If you do not know what products are installed in an instance of Tivoli Workload Automation, perform the following procedure to discover what products are installed:

Open a command prompt and change to the following directory on the Tivoli Workload Scheduler DVD for the operating system of the computer (if you have copied it to hard disk, go to that location): `drive/operating_system/CLI`

Run the following to initialize the Tivoli Configuration Manager environment:

```
swd_env.bat/.sh
```

Run the following software distribution command to verify which software packages have been installed:

```
wdlssp
```

A list of the software packages is displayed, similar to the following:
The details of the packages in the list will depend on which packages have been installed on this computer. In this case, on a Windows computer, an installation for a <TWS_user> called <TWS_user> has been made of the software package blocks for the Tivoli Workload Scheduler scheduling engine and the scheduling engine National Language Support (LP = Language Pack). The value of State depends on whether the package has yet been "committed".

Run the following to remove a software package:

```
wdrmvsp -f <package_name>.<package_version>
```

This command does not remove the log files and configuration files used by Tivoli Configuration Manager. These remain either within the Tivoli Workload Scheduler installation directory, or the system temporary directory.

**Note:** If you experience any problem running these commands, or to understand more about Tivoli Configuration Manager, consult its documentation, which can be found online at [http://publib.boulder.ibm.com/tividd/td/ConfigurationManager4.2.3.html](http://publib.boulder.ibm.com/tividd/td/ConfigurationManager4.2.3.html).

---

**Files backed up during upgrade of Tivoli Workload Scheduler**

During an upgrade from V8.3 and V8.4, some files will be backed up into a .bk file. Files upgraded from V8.5 and higher are not backed up into a .bk file. Additionally, any customized files are not backed up. For example, the file `tws_env.cmd` will be backed up into the file `tws_env.cmd.bk`. These backed up files are replaced with new files. The following is a list of files backed up during the upgrade.

**Windows operating system:**

- `tws_env.cmd`
- `jobmanrc.cmd`
- `TWSCCLog.properties` (this file is not replaced)
- `Startup.cmd`
- `JnextPlan.cmd`
- `MakePlan.cmd`
- `SwitchPlan.cmd`
- `CreatePostReports.cm`
- `UpdateStats.cmd`
- `ResetPlan.cmd`
- `Sfinal`

**UNIX**
When you create a DB2 tablespace with a relative path, the path is constructed in the following way:

DFTDBPATH\DB2_instance\NODE0000\SQLnnnnn\TABLESPACE_REL_PATH

where:

**DFTDBPATH**
For Windows operating system, this is the drive where the DB2 instance is installed. For UNIX and Linux operating systems, this is the home instance of the DB2 installation.

**DB2_instance**
Is the name of the DB2 instance.

**NODE0000**
Is the directory where DB2 database instances are located.

**SQLnnnnn**
Is an incremental directory path that depends on the number of database instances.

**TABLESPACE_REL_PATH**
Is the relative path you specified for the tablespace.

For more information about tablespace relative paths, see the DB2 documentation set.
Notices

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

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

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

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

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

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

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

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

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.
IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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

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

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

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

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information 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 the names and addresses used by an actual business enterprise is entirely coincidental.

### Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml.

Adobe, Acrobat, PostScript and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

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

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

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

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

Special characters
@ character in install directory name, causing the Dynamic Workload Console uninstallation to fail 281

Numerics
4.2-SWDGW-F1P1 fix pack, compatibility problem 214
4.2-TCM-FP02 fix pack, to solve compatibility problem 214
56b - system error given during installation 197

A
access method
silent installation 108
accessibility ix
account creation
Windows 197
Windows 2000 197
add
language packs
command line client 167
new feature
administration HTTP transport field 167
administration HTTPS transport field 167
bootstrap field 167
CSlv2 Client Authentication Listener field 167
CSlv2 Server Authentication Listener field 167
HTTP transport field 166
HTTPS transport field 166
ORB Listener field 167
RMI field 167
SAS Server Authentication Listener field 167
SOAP connector field 167
TWSUser password field 166
option to add the dynamic workload broker resource command with wdinstsp 102
option to add the Java runtime to run job types with advanced options using wdinstsp 102
standard agent, fault-tolerant agent, or domain manager capability with wdinstsp 102
the Java runtime to run job types with advanced options using wdinstsp 102
add feature installation fails 211
adding
connector 83
language packs 83
plug-ins by using the wizard 107
additional plug-ins
uninstallation procedures 175
uninstalling
log files 175
modified files 175
with wizard 175
administration HTTP transport field
new feature
add 167
administration HTTPS transport field
new feature
add 167
agent 89
for distributed environment 79, 89
for end-to-end environment 79, 89
how to uninstall manually 236
install 78
installation
agent display name 81
dynamic workload broker host name 81
dynamic workload broker HTTPS port number 81
host name or IP address 81
installation directory 82
JobManager port 81
user name 80
static environment 3
upgrading dynamic agent 153
upgrading standard or fault-tolerant agent 153
upgrading using a silent installation 145
upgrading using Software Distribution 151
upgrading using the installation wizard 143
upgrading using twsinst 146
agent display name
installation
agent 81
dynamic domain manager 75
master domain manager or backup master 62
agent dynamic
on IBM i 89
agent fault-tolerant
static environment 3
agent installation
unattended, troubleshooting 214
agent uninstalling
silent 172
twsinst 173
wizard 171
Agent, registry attribute 291
AIX
installation problems 199
InstallShield wizard installation fails 210
APARs 198
IY50574 235
APARs (continued)
IY79105 262
application job plug-ins
option to add runtime for Java runtime to run job types with advanced options 148
option to add the Java runtime to run job types with advanced options 153
option to add the Java runtime to run job types with advanced options using wdinstsp 102, 155
application server
credentials problem when installing on Windows 196
installation fails on Windows 2003 domain with credentials problem 206
installation log files 33, 249
installation problem 205
profile creation fails 205
applications
workload environment integrated with 13
attributes, registry file 291
authentication
upgrading 118
authentication mechanism
Tivoli Dynamic Workload Console updating 266
automatically generate ports
installation
dynamic domain manager 76
available functions
for Dynamic Workload Broker 227, 258
for Tivoli Workload Scheduler 227, 258
AWSDEQ024E received in commit step 195
AWSFA035E received 214
AWSFA037E received 208
AWSGAB005W received 196
AWSGAB566E received 215

B
backup
directory 211
backup domain manager configuring 162
static environment 3
backup dynamic domain manager configuring 163
environment 5
installing 72
backup master
installation
dynamic workload broker HTTPS port number 63
backup master domain manager
configuring 161
environment 5
installation
name 62
installing 60
static environment 3
backup master or master domain manager
installation
agent display name 62
company 62
Dynamic workload broker host name 63
host name or IP address 62
JobManager port 62
Netman port 62
password 61
this workstation name 62
user name 61
backup package creation failed 211
batchman
checking if active 239
bc, utility required by InstallShield wizard on Linux 203
before installing
creating database tables 36
before upgrading
upgrading database tables 36
bootstrap field
new feature
add 167
BOOTSTRAP_ADDRESS response file property 303

C
capability
domain manager 3
dynamic agent 5
dynamic domain manager 5
extended agent 3, 6
fault-tolerant agent 3
CDs
Dynamic Workload Console installation 248
checking
prerequisites 24, 114
checkJobsLoop.wait response file property 292
checkPrerequisites.stopOnCheckPrereq response file property 293
CLI
parameter, -installRoot, is invalid, error given on Oracle Solaris 202
wdinstsp 100
wimpspo 100
CMW3202E received 195
command
setup.bin 107
uninstaller.bin 175, 176
command line client (continued)
installation (continued)
remote port 83
user name 83
upgrading 155
commands
wdinstsp 100
wdinstsp agent installation 100
wdinstsp CIT installation 100, 152
wdinstsp to add standard agent, fault-tolerant agent, or domain manager capability 102
wdinstsp to add standard agent, fault-tolerant or domain manager capabilities 153
wdinstsp to add the Java runtime to run job types with advanced options 102, 155
wimpspo 100
commands and scripts
ln
to link directories to the .swdis directory 207, 212
makesec
create Security file 236
ps, used before manual uninstallation 239
shut, used before manual uninstallation 239
stop
used before manual uninstallation 239
twsinst, files not being correctly copied before running 208, 216
unlink
used before manual uninstallation 239
wconvcat 214
wdlssp, used before manual uninstallation 239
wdrmvsp, used before manual uninstallation 239
wlsinst 214
commit step fails 215
company
installation
fault-tolerant agent 81
master domain manager or backup master 62
COMPANY_NAME property customize
DB2 39
ORACLE 48
configuring
backup domain manager 162
backup dynamic domain manager 163
backup master domain manager 161
domain manager 161
dynamic agent 165
dynamic domain manager 163
dynamic scheduling after installation 168
dynamic scheduling after upgrade 168
fault-tolerant agent 164
master domain manager 160
connection
to Tivoli Workload Scheduler 228, 259
connectors
adding 83, 166
uninstalling manually on Windows 242
connectors
uninstall manually 241
console
portfolio 262
start 262
conventions used in publications ix
cpuCfgPanel.addFINAL response file property 293
cpuCfgPanel.company response file property 293
cpuCfgPanel.jmPort NumberHttps response file property 293
cpuCfgPanel.masterCPU response file property 293
cpuCfgPanel.tcp PortNumber response file property 293
cpuCfgPanel.tdvbHostName response file property 293
cpuCfgPanel.thisCPU response file property 293
cpuCfgPanel.tws PortNumber response file property 293
create
SQL files before installing a dynamic domain manager
Oracle 52
SQL files before installing a master domain manager
Oracle 42, 49
CREATE_WAS_SERVICE response file property 203
creating database tables before installing 36
before upgrading 36
creating database tables DB2 property file
COMPANY_NAME property 39
DB_USER property 37
EIP_PORT property 39
HOST_NAME property 39
TWS_DATA_TS_PATH property 38
TWS_DB property 38
TWS_LOG_TS_NAME property 38
TWS_TS_NAME property 38
TWS_USER property 38
WAS_SEC_PORT property 39
credentials problem causing installation of application server on Windows 2003 domain to fail 206
credentials problem for application server when installing on Windows 196
Cslv2 Client Authentication Listener field
new feature
add 167
CSIV2 Server Authentication Listener
field
new feature
add 167
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS response file
property 303
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS response file
property 303
customize
DB2
properties file 37
Oracle
properties file 47
customizeDB2SQL.properties file
customize
DB2 37
customizeORACLESQl.properties file
customize
ORACLE 47
customizeSQL
generating SQL files
DB2 40

DB2 (continued)
run before installing dynamic domain manager
SQL files 42
run before upgrading Tivoli Workload Scheduler
SQL files 43, 45
upgrading database tables
before upgrading 37
db2CheckPrereqs.db2Directory.property file property 293
db2CheckPrereqs.db2RemoteNode.property file property 294
db2CheckPrereqs.db2RemotePort.property file property 294
db2CheckPrereqs.db2TwsDBPwd.property file property 294
db2CheckPrereqs.db2TwsDBUser.property file property 294
db2CheckPrereqs.db2AdminPwd.property file property 294
db2CheckPrereqs.db2AdminUser.property file property 294
db2CheckPrereqs.db2LocalAdminPwd.property file property 294
db2CheckPrereqs.db2LocalAdminUser.property file property 294
db2CheckPrereqs.db2ServerCfg.instanceName.property file property 295
db2CheckPrereqs.db2ServerCfg.db2AdminPwd.property file property 295
db2CheckPrereqs.db2ServerCfg.db2AdminUser.property file property 295
db2CheckPrereqs.db2ServerCfg.db2InstancePort.property file property 295
db2CheckPrereqs.db2ServerCfg.db2ServerPort.property file property 295
DCS_UNICAST_ADDRESS response file property 303
deleting files
too slow after manual uninstall 240
directories
backup 211
swdis 211
work 211
directory for SSL files
upgrade master domain manager 126
disconnected command line, of Tivoli Configuration Manager 214
disk space
not enough for installation 216
display name
agent
master domain manager or backup master 62
installation
agent 81
dynamic domain manager 75
disse0005E received 211
disse0006E received 214
disse0028E received 211
disse0024E received 211
distributed workload
environment 7
environment with dynamic scheduling capabilities 9
environment with static and dynamic scheduling capabilities 11
distributed-driven
workload environment for z/OS 14
domain
amount of network traffic 17
dependencies between jobs 17
firewalls 17
internetwork dependencies 18
level of fault-tolerance required 17
localized processing 16
number of geographic locations 16
number of workstations, applications, and jobs 16
planning 15, 16
SSL or GSKit 17
system performance and other criteria 17
time zones 16, 17
topology
multiple 19
single 17
types of applications 17
Windows network 17
domain manager
configuring 161
static environment 3
upgrading 153
upgrading using the installation wizard 143
domain name
installation
dynamic domain manager 74
DVs
installation 26
dynamic agent
capability 5
configuring 165
environment 5
installation
user name 80
installing 79
maintaining ID 215
maintaining name 215
multiple installations 215
on IBM i 89
reinstalling 215
repeated installations 215
retrieving ID 215
several installations 215
dynamic and static scheduling capabilities
environment with 11
dynamic domain manager
configuring 163
environment 5
installation
agent display name 75
automatically generate ports 76
domain name 74
dynamic workload broker host name 75
dynamic workload broker netman port 78
dynamic workload broker workstation name 78
host name or IP address 75
JobManager port 75
master domain manager name 74
dynamic domain manager (continued)
installation (continued)
master domain manager netman
port 72
Netman port 75
password 74
this workstation name 74
user name 73
installation fails 209
installation fails after database
configuration 209
installing 72
run SQL files before installing
DB2 42
Oracle 52
uninstalling 171
dynamic scheduling
enabling 83, 158
enabling after installation 168
enabling after upgrade 168
dynamic scheduling capabilities
environment with 9
dynamic workload broker
installation
dynamic workload broker HTTPS
port number 75
Dynamic Workload Broker
available functions 227, 258
server connection 229, 260
dynamic workload broker host name
installation
agent 81
dynamic domain manager 75
Dynamic workload broker host name
installation
master domain manager or backup
master 63
dynamic workload broker HTTPS port
number
installation
agent 81
dynamic domain manager 75
master domain manager or backup
master 63
dynamic workload broker workstation
name
installation
dynamic domain manager 78
master domain manager 71
Dynamic Workload Console
accessibility ix
configuration 262
for Tivoli Workload Scheduler
Version 8.3 Fix Pack 3 225, 256
connection
to Dynamic Workload Broker
components 229, 260
getting started 262
installation
advanced 221, 252
CDs 248
default 221, 251
images 248
log files 248
methods 246
on embedded WebSphere
Application Server 220, 251
Dynamic Workload Console (continued)
installation (continued)
on existing external WebSphere
Application Server 222, 253
on existing instance of the
embedded WebSphere
Application Server 222, 253
tsate scenarios 246
setup file 248
silent 223, 254
types 220, 251
using launchpad 219, 250
using response file 223, 254
using wizard 219, 250
installation and uninstallation log
files 271
log files 248
overview 245
remove
manually 272, 273
starting and stopping 229, 260
troubleshooting 271
uninstall 270
clean-up 272, 273
in silent mode 270
manually 272, 273
using wizard 270
updating
authentication mechanism 266
upgrade
failed, recovering 272
upgrading
overview 266
silently 268
upgrading on embedded
WebSphere Application
Server 267
using launchpad 268
using wizard 268
user interface 226, 257
Dynamic Workload Console problems
with
uninstallation 280
upgrade 280
E
education x
EI_PORT property
customize
DB2 39
ORACLE 48
ENABLE_TDWB response file
property 304
ENABLE_TWS response file
property 304
enabling
dynamic scheduling 83, 158
dynamic scheduling after
installation 168
dynamic scheduling after
upgrade 168
distributed workload environment
planning 12
Dynamic Workload Console problems
with
environment 280
upgrade 280
E 320
IBM Tivoli Workload Scheduler: Planning and Installation
file (continued)

tws4kapps_uninstall.log silent uninstallation 176
tws4kapps_uninstall.log uninstallation 175
TWSAPPS_RespFile_UNIX.txt silent installation 108
TWSAPPS_RespFile_windows.txt silent installation 108
file names, case changed during copy 216
file system structure
upgrade master domain manager 125
files
/etc/password 150
case changed during copy 216
FINAL 160
logs
installation 179
uninstallation 179
names, case changed during copy 216
not correctly copied before running twsinst 208, 216
Security
checking existence of 236
swdis.ini 100
Symphony 20
twsinst, before running not being correctly copied 208, 216
TWSRegistry.dat 239, 291
files changed
upgrade 113
FINAL
adding 160
final job stream
adding 160
firewall stopping installation of the Dynamic Workload Console 276
fix packs
4.2-SWGW-F1P1, compatibility problem 214
4.2-TCM-FP02, to solve compatibility problem 214
installing using Tivoli Configuration Manager
incompatibility problem 214
insufficient disk space 211
silent installation fails 207, 208
folders changed
upgrade 113

G

gateway component of software distribution, incompatibility 214

I

IBM i
agent dynamic 89
images
Dynamic Workload Console installation 248
images_dir
generating SQL files
DB2 39, 40
install
agent 78

INDEX

INSTALL_METHOD response file property
property 304
InstallAnywhere error messages messages 191
installation
adding connector 166
adding features 83, 165
additional method 107
agent
agent display name 81
dynamic workload broker host name 81
dynamic workload broker HTTPS port number 81
host name or IP address 81
installation directory 82
JobManager port 81
user name 80
AIX, problems 199
application server installation problems 205
backup master domain manager workstation name 62
checking prerequisites 24
command line client 82
password 83
remote host 83
remote port 83
user name 83
correcting a failed step and continuing 186
DVs 26
dynamic domain manager
agent display name 75
automatically generate ports 76
domain name 74
dynamic workload broker host name 75
dynamic workload broker HTTPS port number 75
dynamic workload broker netman port 78
dynamic workload broker workstation name 78
host name or IP address 75
JobManager port 75
master domain manager name 74
Netman port 75
password 74
this workstation name 74
user name 73
Dynamic Workload Console CDs 248
images 248
in silent mode 247
methods 246
on embedded WebSphere Application Server 220, 251
on existing external WebSphere Application Server 223, 253
on existing instance of the embedded WebSphere Application Server 222, 253
sample scenarios 246
setup file 248
types 220, 251
using launchpad 246

Index 321
installation (continued)
Dynamic Workload Console (continued)
using wizard 246
Dynamic Workload Console log files 271
example procedure for problem resolving 189
fails
"Error writing file = " 207
"Error writing file = 28" 208
application server profile creation fails 205
AWSDEQ024E error received 195
AWSFAB037E error received 211
CMW3202E error received 209
dynamic domain manager 209
host name truncated 195
InstallShield wizard on
Windows 208
master domain manager 208
miscellaneous 216
NoClassDefFoundError error 210
Oracle Solaris with error
"command line parameter,
-installRoot, is invalid" 202
recovery using wizard 179
software package block 211, 212
UNIX with JVM validation
error 200
with error AWSFAB035E 214
with error AWSGAB566E 215
with error DISSE032E 211
with installation image problem on
NFS mount 210
fails on Red Hat Enterprise Linux
6.0 204
fault-tolerant agent
company 81
master domain manager name 81
Netman port 81
workstation name 81
from shared folder fails on
Windows 277
hangs (Dynamic Workload
Console) 276
hangs on AIX 199
hangs on AIX 6100-05 199
HP-UX, problems 201
InstallShield wizard
"Add feature" installation
fails 211
does not start on HP-UX 201
fails on HP-UX with a "run
error" 201
JRE does not install on
HP-UX 201
JRE does not install on Linux 203
Linux
start of Tivoli Workload Scheduler
gives errors 203
Linux, problems 203
log files 32, 179
log files, DB2 33
installing (continued)
Java runtime 79
language packs with Software Distribution 103
master domain manager 60
software package block 151
syntax CIT installation using wdinstsp 152
installing additional plug-ins before 106
installLocation response file property 297, 308
installRoot not valid on Oracle Solaris 202
InstallShield wizard
"Add feature" installation fails 211
cannot install JRE on HP-UX 201
cannot install JRE on Linux 203
does not start on HP-UX 201
fails on AIX or HP-UX with a "run error" 210
fails with "run error" 210
fails with a "run error" on HP-UX 201
installation and uninstallation log files 32, 248
problem using with the Dynamic Workload Console 275
recovering installation with 179
uninstallation fails on Windows 232
InstallShield wizard backup master domain manager installation localopts 217
InstallShield wizard installation backup master domain manager localopts 217
fails
AWSJSI145E 217
InstallShield wizard Oracle installation fails
AWSJSI145E 217
INSTANCE_PATH response file property 296
Integration Workbench
installing 308, 309
installing using the remote Eclipse Site 309
upgrading 309
upgrading installed as a plug-in 310
interactive wizard
problem using with the Dynamic Workload Console 275
interface
command line client 6
dynamic workload broker command line 6
Dynamic Workload Console 6
Job Brokering Definition Console 6
master domain manager command line 6
internetwork dependencies domain 18
IP address installation agent 81
IP address or host name installation
  dynamic domain manager 75
  master domain manager or backup master 62
IPC_CONNECTOR_ADDRESS response file property 304
IS_BACKUP_DIR response file property 305
IS_DESTINATION response file property 305
IS_UPGRADE response file property 305
ISC_ADMIN_FULL_USER response file property 305
ISC_ADMIN_PASSWORD response file property 305
ISC_APPSERVER_DIR response file property 306
ISMP installation method 31
ISMP silent mode installation method 31
IsOnlyFTAConnectorToUninstall. property 297
IY50574, APAR 235
IY52481, APAR 198
Java runtime
installing 79
Java Runtime Environment
cannot be installed on HP-UX 201
cannot be installed on Linux 203
validation problem on UNIX 200
javahome_dir generating SQL files
DB2 40
jobman and JOBMAN checking if active 239
JobManager port installation
agent 81
dynamic domain manager 75
master domain manager or backup master 62
JVM
causing installation to fail on Linux RHEL V5 278
causing installation to fail on Suse Linux 278
causing product installation to fail on Linux RHEL 5 204
causing product installation to fail on Suse Linux 204
jvmtimer, need to use for UNIX installation 200
K
kernel parameters, max_thread_proc 201
keys, registry, Windows, removing 243
language packs
add
command line client 167
adding 83
installing 93, 149
installing with Software Distribution 103
removing 171
uninstalling 174
launchpad
installation 30
problems using with the Dynamic Workload Console 275
LDAP
upgrading 118
Tivoli Dynamic Workload Console 266
license Accepted response file property 308
licenseAccepted response file property, TDWC 306
licenseAccepted response file property, TWS 297
Linux
errorneous warning messages displayed from launchpad 275
installation fails if host name not FQDN 278
installation problems 203
InstallShield wizard installation cannot install JRE 203
InstallShield wizard installation RHEL 5 (x86–64) install or uninstall of the Dynamic Workload Console fails to start 279
RHEL 5 and Suse 11 product installation fails (JVM) 204
RHEL V5 and Suse V11 installation fails (JVM) 278
start of Tivoli Workload Scheduler gives errors after installation 203
Linux user accounts 34
ln, command to link directories to the .swdis directory 207, 212
localized processing
domain 16
localopts
checking default settings 235
Log Analyzer
description 308
log file
tws4apps_ia_uninstall.log silent uninstallation 176
tws4apps_status.log uninstallation 175
tws4kapps_uninstall.log silent uninstallation 176
tws4kapps_uninstall.log uninstallation 175
tws4plugins_install.log 107
log file for uninstalling
tws4apps_ia_uninstall.log 175
log file not written by failed silent installation 204
log file silent installation
tws4kapps_ia_install.log 108
log files 32
DB2 installation 33
Dynamic Workload Console 271
embedded WebSphere Application Server installation 33, 249
installation 179
packaging for support 179
uninstallation 179
uninstalling additional plug-ins 175
log successfully, but Tivoli Integrated Portal installation fails 276
LPList, registry attribute 291
LPName, registry attribute 291
mailman
checking if active 239
MaintenanceVersion, registry attribute 291
MajorVersion, registry attribute 291
makesec
create Security file 236
manual uninstallation
agents 236
connector 241
master domain manager 236
manually
Dynamic Workload Console uninstall 272, 273
master domain manager
configuring 160
environment 4
installation
dynamic workload broker HTTPS port number 63
dynamic workload broker workstation name 71
installation fails 208
checking if active 239
run SQL files before installing
DB2 41
Oracle 49
static environment 2
uninstall manually 236
master domain manager name
installation
dynamic domain manager 74
fault-tolerant agent 81
master domain manager or backup master
installation
agent display name 62
company 62
Dynamic workload broker host name 63
host name or IP address 62
JobManager port 62
Netman port 62
password 61
this workstation name 62
user name 61
master domain manager upgrade directory for SSL files 126
file system structure 125
product directory 124
master, global option 235
max_thread_proc kernel parameter 201
MAXDSIZ, configuration parameter needed for HP-UX 201
MDL_USER property
customize
ORACLE 47
memory
virtual, out of 215
message
InstallAnywhere return code 191
method
for installing 107
methods
for uninstalling 175
MinorVersion, registry attribute 291
modified files
uninstalling additional plug-ins 175
new feature
add
administration HTTP transport field 167
administration HTTPS transport field 167
bootstrap field 167
CSlv2 Client Authentication Listener field 167
CSlv2 Server Authentication Listener field 167
HTTP transport field 166
HTTPS transport field 166
ORB Listener field 167
RMI field 167
SAS Server Authentication Listener field 167
SOAP connector field 167
TWSUser password field 166
NFS mounted installation image, installation fails 210
NoClassDefFoundError error 210
operating system
generating SQL files
DB2 39
options
product response file 109, 177
product silent installation 109
product silent uninstallation 177
Oracle
creating database tables
before installing 36, 47
customize
properties file 47
generating
SQL files 48
run before installing a dynamic domain manager
SQL files 52
run before installing a master domain manager
SQL files 49
run before upgrading
SQL files 57
run before upgrading Tivoli Workload Scheduler
SQL files 53
upgrading database tables
before upgrading 36, 47
ORACLE
customize
COMPANY_NAME property 48
EIF_PORT property 48
HOST_NAME property 48
MDL_USER property 47
TWS_LOG_TS_NAME property 48
TWS_PASSWORD property 47
TWS_TS_NAME property 48
TWS_TS_TEMP_NAME property 48
TWSTEMPDIR property 47
WAS_SEC_PORT property 48
Oracle E-Business Suite applications workload environment integrated with 13
oracleCheckPrereqs. oracleDirectory response file property 297
oracleServerCommunication Info.netServiceName response file property 297
oracleServerCommunication Info.oracleAdminPwd response file property 298
oracleServerCommunicationInfo. oracleAdminUser response file property 298
ORB Listener field new feature
ORB_LISTENER_ADDRESS response file property 306
output tab, in step window 184
overview installation 23
upgrading Dynamic Workload Console 266

P
PackageName, registry attribute 291
page file, too small 215
parameter twsinst update
-addruntime 148
-backup_dir 148
-displayname 148
-domain 148
-hostname 148
-inst_dir 149
-jimport 149
-jimportssl 149
-lang 149
-nobackup_dir 149
-password 149
-reset_perm 149
-skip_usercheck 150
-tdwbbhostname 150
-tdwbport 150
-uname 150
-update 150
-wait 150
password
installation command line client 83
dynamic domain manager 74
master domain manager or backup master 61
password incorrect for TWS_user on UNIX 200
PatchVersion, registry attribute 291
pausing an installation 189
Peoplesoft applications workload environment integrated with 13
planning distributed workload environment 7
distributed workload environment with dynamic scheduling capabilities 9
planning (continued) distributed workload environment with static and dynamic scheduling capabilities 11
distributed-driven workload environment for z/OS 14
domain 15, 16
end-to-end workload environment 12
environment 7, 9, 11
localized processing in your domain 16
workload environment integrated with external systems 13
plug-ins adding with the wizard 107
PLUGINS_TO_UNDEPLOY response file option 178
port dynamic workload broker HTTPS number 75
JobManager 75, 81
master domain manager or backup master 62
Netman fault-tolerant agent 81
master domain manager or backup master 62
port number HTTPS dynamic workload broker installation master domain manager or backup master 63
portfolio console 262
ports WebSphere Application Server 221, 252
ports automatically generate installation dynamic domain manager 76
post installation configuring a backup domain manager 162
configuring backup dynamic domain manager 163
configuring backup master domain manager 161
configuring domain manager 161
configuring dynamic agent 165
configuring dynamic domain manager 163
configuring fault-tolerant agent 164
configuring master domain manager 160
post-installation actions 212
pre-installation actions 212
preparing installation 23
prerequisites checking 24, 114
problem scenarios, installation 193
procedure for uninstalling 175
procedure for resolving installation problems 189
product before installing additional plug-ins 106
silent installation 108
silent uninstallation 176
uninstalling with wizard 175
product directory upgrade master domain manager 124
ProductID, registry attribute 291
profile, WebSphere Application Server, different, causing the Dynamic Workload Console installation to fail 279
properties file DB2 customize 37
Oracle customize 47
properties tab, in step window 183
property file DB2 creating database tables
COMPANY_NAME property 39
DB_USER property 37
EIF_PORT property 39
HOST_NAME property 39
TWS_DATA_TS_PATH property 38
TWS_DB property 38
TWS_LOG_TS_NAME property 38
TWS_LOG_TS_PATH property 39
TWS_TS_NAME property 38
TWS_USER property 38
WAS_SEC_PORT property 39
property_file_path generating SQL files DB2 40
ps, command used before manual uninstallation 239
publications ix

R
recovInstReg.run response file property 298
Red Hat Enterprise Linux 6.0 failed installation 204
Red Hat Enterprise Linux V5 product installation failing on (JVM) 204
Red Hat Enterprise Linux V5, (x86–64), install or uninstall of the Dynamic Workload Console failing on 279
Red Hat Enterprise Linux V5, installation failing on (JVM) 278
registry attribute 291
registry entries, deleting manually UNIX 239
Windows 237, 243
registry file attributes 291
example 291
recreating 156
upgrading with corrupt files 156
Relational database management systems installation 29
remote host
command line client
installation 83
remote port
installation 57
command line client 83
remove
Dynamic Workload Console manually 272, 273
removing the product 170
dynamic domain manager 171
silent 172
twsinst 173
wizard 171
rerun installation wizard, or resume 187
response file
product install options 109
product uninstall options 177
TWSAPPS_RespFile_windows.txt silent installation 108
response file install option
TWSAPPS_PLUGIN_FILE_NAME 109
USER_INSTALL_DIR 109
response file missing, causing silent installation to fail 280
response file option
PLUGINS_TO_UNDEPLOY 178
response file uninstall option
USER_INSTALL_DIR 177
response files
silent installation 84
template 87
REST_NOTIFICATION_ADDRESS response file property 306
response file property 306
response file uninstall
setup file
script webui 225, 256
SDK_ECLIPSE_BUNDLED response file property 298
SDK_UPDATESITE response file property 298
security implications of installation 234
Security, file
checking existence of 236
selectRDBMSPanel.dbmsSelected response file property 298
silent, command, used before manual installation 84
setup.bin
closure panel before using InstallShield wizard 194, 232
deleting 237
fail to start after installation 198
Service Control Manager error 232
setup file
Dynamic Workload Console installation 248
silent, command for installation 107
shared Windows folder, installation fails from 277
shut, command, used before manual uninstallation 239
Silent
uninstalling 172
silent installation 84
Dynamic Workload Console 223, 254
fails without writing a log 204
of the Dynamic Workload Console 247
response file template 87
Tivoli Workload Scheduler for Additional Plug-ins install options 109
troubleshooting 204
silent installation log file
TWS4APPS_LA_INSTALL_LOG 108
TWS4APPS_INSTALLED_LOG 108
silent installation of the Dynamic Workload Console problems with 279
silent installation wizard
fails with "Error writing file = " 207
silent installation wizard (continued)
fails with "Error writing file =" 208
recovering 190
silent mode ISMP
Installation method 31
silent uninstall
of the Dynamic Workload Console 270
silent uninstallation
additional plug-ins uninstall options 177
Tivoli Workload Scheduler for Additional Plug-ins 176
SOAP connector field
new feature
add 167
SOAP_CONNECTOR_ADDRESS response file property 306
Software Distribution
installing language packs 103
software package block
creating and installing 151
software package block installation and uninstallation log files 33
Solaris
installation fails with error "command line parameter, -installRoot, is invalid" 202
installation problems 202
SQL files
generating
DB2 39
Oracle 48
run before installing a dynamic domain manager
Oracle 52
run before installing a master domain manager
DB2 41
Oracle 49
run before installing dynamic domain manager
DB2 42
Oracle 57
run before upgrading
Oracle 57
run before upgrading Tivoli Workload Scheduler
DB2 43, 45
Oracle 53
SSL files directory
upgrade master domain manager 126
stageman
checking if active 239
standard agent
capability static 3
environment static 3
starting
console 262
Dynamic Workload Console 229, 260
server 229, 260
static and dynamic scheduling
capabilities
environment with 11
static capability
fault-tolerant agent 3
static capability (continued)
standard agent 3
static network
backup master domain manager 3
domain manager 3
master domain manager 1
status tab, in step window 182
step
configuring a backup domain manager 162
configuring backup dynamic domain manager 161
configuring backup master domain manager 161
configuring domain manager 161
configuring dynamic agent 165
configuring dynamic domain manager 163
configuring fault-tolerant agent 164
configuring master domain manager 160
step list
status 182
using 181
step window
using 182
step, installation, failed, correcting and continuing 186
stepped installation wizard 179
stop, command
used before manual uninstallation 239
stopping an installation 189
Dynamic Workload Console 229, 260
server 229, 260
services 142
stopWas command hangs during install of the Dynamic Workload Console 276
structure
installation DVD structures 26
summary.log, file 179
support
packaging log files for 179
Suse Linux 11
product installation failing on (JVM) 204
Suse Linux V11, installation failing on (JVM) 278
swap space, out of 215
swd_env
wizard, troubleshooting 214
SWDGW component of Tivoli Configuration Manager, incompatibility 214
swds, directory
insufficient space 211
Symphony file 20
syntax
wdinstsp to add the Java runtime to run job types with advanced options 102, 155
syntax agent installation wdinstsp 100
syntax CIT installation wdinstsp 100, 152
T
Tdwb Config, tdwbCpuName 297, 298
TdwbConfig, tdwbCpuPort 297, 299
techical training x
this workstation name installation
dynamic domain manager 74
master domain manager or backup master 62
time zone
overview 21
Tivoli Configuration Manager
incompatibility problem with fix packs 214
installation fails 211, 212
insufficient disk space 211
Tivoli Dynamic Workload Console
configuration 262
getting started 262
overview 245
starting and stopping 229, 260
troubleshooting 271
uninstall 270
updating
authentication mechanism 266
upgrading
overview 266
Tivoli Integrated Portal
installation
from the DVD or elmages 224, 255
Tivoli Netman for TWS_user, deleting service 237
tivoli technical training x
Tivoli Token Service
fails to start after installation 198
for TWS_user, deleting service 237
Tivoli Workload Scheduler 245
available functions 227, 258
eengine connection 228, 259
Tivoli Workload Scheduler agents 88
Tivoli Workload Scheduler agents uninstalling
silent 172
twinst 173
wizard 171
Tivoli Workload Scheduler for Additional Plug-ins
before installing 106
installation
additional plug-in 107
uninstalling
with silent uninstallation 176
Tivoli Workload Scheduler for Additional Plug-ins installation fails
AWSJS145E 217
Tivoli Workload Scheduler for Additional Plug-ins temp installation fails
does not have enough space 217
Tivoli Workload Scheduler for Applications 245
Tivoli Workload Scheduler for z/OS 245
Tivoli Workload Scheduler service for TWS_user
deleting 237
fails to start after installation 198
Tivoli Workload Scheduler Version 8.3 Fix Pack 3
configuration for Dynamic Workload Console 225, 256
tools
Integration Workbench 308
training
technical x
troubleshooting
application server installation problems 205
fix pack installation 234
installation 178, 194
AIX 199
HP-UX 201
Oracle Solaris 202
UNIX 200
installation scenarios
Dynamic Workload Console 275
installations
Linux 203
miscellaneous installation problems 206
uninstallation 232
upgrading 218
truncated host name on Windows: installation fails 195
TWA_INSTANCE_PATH response file property 306
TWS_DATA_TS_PATH property
customize
DB2 38
TWS_DB property
customize
DB2 38
TWS_LOG_TS_NAME property
customize
DB2 38
ORACLE 47
TWS_LOG_TS_PATH property
customize
DB2 39
TWS_PASSWORD property
customize
ORACLE 44
TWS_TS_NAME property
customize
DB2 38
ORACLE 48
IBM Tivoli Workload Scheduler: Planning and Installation

TWS\_TS\_TEMP\_NAME property customize
ORACLE 48
TWS\_USER property customize
DB2 38
ORACLE 47
tws4apps\_ia\_install.log
log file silent installation 108
tws4apps\_ia\_uninstall.log
log file for silent uninstallation 176
log file for uninstalling 175
tws4apps\_install.log
log file for silent installation 108
tws4apps\_uninstall.log
log file for silent uninstallation 176
log file for uninstallation 175
tws4plugins\_install.log
log file installation 107
TWSAPPS\_PLUGIN\_FILE\_NAME
zip file 109
TWSAPPS\_RespFile\_UNIX.txt 108
TWSAPPS\_RespFile\_Windows
installation response file for product 108
twsClCiGPanel\_password response file property 299
twsClCiGPanel\_remoteHost response file property 299
twsClCiGPanel\_remotePort response file property 299
twsClCiGPanel\_user response file property 299
TWSCLILanguagesPanel\.* response file properties 299
twsCLILocationPanel\_directory response file property 299
TWSConnRegistry.dat 241
twsDBCfg\_dbName response file property 299
twsDBCfg\_report TablespaceName
response file property 300
twsDBCfg\_report TablespacePath response file property 300
twsDBCfg\_tableSpaceName response file property 300
twsDBCfg\_tableSpacePath response file property 300
twsinst 88, 89
fails "The twsinst script is being run from the wrong directory." 208
miscellaneous 216
with error AWSFAB035E 214
with error AWSGAB566E 215
files not being correctly copied before running 208, 216
installation and uninstallation log files 32
installation method 31, 88
unattended, troubleshooting 205
uninstalling 173
UNIX usage 147
Windows usage 148
twsinst installation agent
swd\_env 214
twsismp\_log, file 179
twsLocationPanel\_directory response file property 300
twsLocationPanel\_symLinkOption
response file property 300
twsOracleDBCfg\_isPartitioned response file property 300
twsOracleDBCfg\_twsDBPwD response file property 300
twsOracleDBCfg\_twsDBUser response file property 300
twsOracleDBCfg\_tws DataTablespace
response file property 300
twsOracleDBCfg\_tws TempTablespace
response file property 300
twsOracleDBCfg\_tws ReportTablespace
response file property 300
twsp\_MySQL\_Admin\_Manager\_Install\_Wizard\_Install\_Agent\_log 241
TWSConnRegistry.dat 241
twsPortsPanel\_portSrvAuth response file
wtsPortsPanel\_portSOAP response file
twsPortsPanel\_portSAS response file
twsPortsPanel\_portRMI response file
twsPortsPanel\_portORB response file
twsPortsPanel\_portHTTP response file
twsPortsPanel\_portEif response file
wtsPortsPanel\_portAdminSec response property 301
wtsPortsPanel\_portAdmin response file property 301
wtsPortsPanel\_portHTTPS response file property 301
wtsPortsPanel\_portMtlsAuth response file property 301
wtsPortsPanel\_portORB response file property 301
wtsPortsPanel\_portRMI response file property 301
wtsPortsPanel\_portSAS response file property 301
wtsPortsPanel\_portSOAP response file property 301
wtsPortsPanel\_portSrvAuth response file property 301
TWSRegistry.dat, file 239, 291
TWS\_TEMP\_DIR property
DB2 37
ORACLE 47
VtwSUpgradePanel\_backupOldInstance
response file property 301
VtwSUpgradePanel\_bkcpDirectory response file property 302
VtwSUpgradePanel\_bkcpProfileDirectory
response file property 302
VtwSUpgradePanel\_dumpDirectory
response file property 302
TWSUser
account creation
Windows 197
Windows 2000 197
deleting from registry UNIX 239
Windows 237
TWSUser password field
new feature add 166
TWSZConnRegistry.dat 241
UNIX (continued)
password incorrect for TWS_user 200
silent fix pack installation fails 207, 208
uninstalling manually 239
UNIX user accounts 34
unlink workstation 142
unlink, command used before manual uninstallation 239
update of the embedded WebSphere Application Server fails during the fix pack installation 234
UPDATE_INSTALLER_DIR response file property 306
updating Tivoli Dynamic Workload Console authentication mechanism 266
upgrade checking prerequisites 114
existing software versions 112
files and folders changed 113
of the Dynamic Workload Console, problems with 280
problems 218, 234
recovering 191
safe 121
troubleshooting 178
upgrade master domain manager directory for SSL files 126
file system structure 125
product directory 124
upgrading adding features 158
agent 141
agent using a silent installation 145
agent using Software Distribution 151
agent using the installation wizard 143
agent using twsinst 146
authentication 118
Tivoli Dynamic Workload Console 266
command line client 155
domain manager 153
domain manager using the installation wizard 143
domain managers 141
dynamic agent 153
Dynamic Workload Console on embedded WebSphere Application Server 267
overview 266
fault-tolerant agent 156
Integration Workbench 309
Integration Workbench installed as a plug-in 310
LDAP 118
standard or fault-tolerant agent 153
syntax CIT installation using wdinstsp 152
with corrupt registry files 156
user name creating 99

V
variables language
Brazilian Portuguese 104
Chinese, Simplified 104
Chinese, Traditional 104
French 104
German 104
install_dir 104
Italian 104
Japanese 104
Korean 104
Spanish 104
tws_user 104
Software Package Block backup 97
customize
company 97
domain 98
fresh_install 98
from_release 98
install_dir 98
jm_port 98
jm_sec_port 98
master_cpu 98
password 98
pwd 98
tcp_port 98
tdwb_hostname 99
variables (continued)
Software Package Block (continued)
tdwb_port 99
this_cpu 99
tws_user 99
upgrade 99
symlink
TWA/TWS/bin/at 26
TWA/TWS/bin/batch 26
TWA/TWS/bin/datecalc 26
TWA/TWS/bin/jobstdl 26
TWA/TWS/bin/maestro 26
TWA/TWS/bin/morestdl 26
TWA/TWS/bin/muser 26
TWA/TWS/bin/parms 26
verifying Tivoli Workload Scheduler installation 235
version 8.3
run SQL files before upgrading Tivoli Workload Scheduler
DB2 43
Oracle 53
version 8.4
run SQL files before upgrading Tivoli Workload Scheduler
DB2 43
Oracle 53
version 8.5
run SQL files before upgrading Tivoli Workload Scheduler
DB2 43
Oracle 53
version 8.5.1
run SQL files before upgrading Tivoli Workload Scheduler
DB2 45
Oracle 57
version 8.5.1.1
run SQL files before upgrading Tivoli Workload Scheduler
DB2 45
Oracle 57
virtual memory, out of 215

W
WAS_CELL_NAME response file property 307
WAS_NODE_NAME response file property 307
WAS_PROFILE_NAME response file property 307
WAS_SEC_PORT property customize
DB2 39
ORACLE 48
WAS_SERVER_NAME response file property 307
WC_adminhost response file property 307
WC_adminhost_secure response file property 307
WC_defaulthost response file property 307
WC_defaulthost_secure response file property 307
Index 329
wconvcat, command 214
wdinstsp
  syntax agent installation 100
  syntax CIT installation 100, 152
  syntax to add standard agent, fault-tolerant agent, or domain manager capability 102
  syntax to add standard agent, fault-tolerant or domain manager capabilities 153
  syntax to add the Java runtime to run job types with advanced options 102, 155
wdlssp, command used before manual uninstallation 239
wdrmvsp, command used before manual uninstallation 239
WebSphere Application Server
  choosing instance 219, 250
  ports 221, 252
WebSphere Application Server, installation of the Dynamic Workload Console fails when installing on different profile 279
webui script 225, 256
Windows
  2003 domain, application server installation fails with credentials problem 206
  credentials problem for installing on application server 196
  file deletion to slow after manual uninstallation 240
  installation receives warning AWSGAB005W 196
  with InstallShield wizard, fails 194
  installation fails
    host name truncated 195
  installation of the Dynamic Workload Console fails on different external WebSphere Application Server profile 279
  installation problems 194
  registry keys, removing 243
  shared folder, installation fails from 277
  undefined error message displayed from launchpad 275
  uninstallation fails because the embedded WebSphere Application Server not stopped 232
  uninstallation fails in restore profiles step 232
  uninstallation with InstallShield wizard fails 232
  uninstalling manually 237
  uninstalling the connector manually on 242
  wizard (continued)
    uninstalling additional plug-ins 175
  wlsinst, command 214
  work, directory 211
  workstation unlinking 142
  workstation class definition 21
  workstation name installation
    backup master domain manager 62
    fault-tolerant agent 81
  workstation name dynamic workload broker installation
    dynamic domain manager 78
    master domain manager 71
  writer checking if active 239

Z
z/OS applications
  workload environment integrated with 13