

IBM Workload Scheduler

Upgrading from V8.5.1 to V9.3

IBM

IBM Workload Scheduler

Upgrading from V8.5.1 to V9.3

IBM

Contents

Chapter 1. Steps to upgrade from V8.5.1 to V9.3 1

Chapter 2. Upgrading IBM Workload Scheduler 3

Upgrading from V8.5.1	3
Upgrading the fault-tolerant agent and the distributed connector	5
Upgrading the fault-tolerant agent and the Dynamic Workload Console	5
Upgrading the fault-tolerant agent, the Dynamic Workload Console, and the z/OS connector	5
Upgrading the fault-tolerant agent and the z/OS connector	6
Upgrading the master domain manager and the Dynamic Workload Console	6
Upgrading the master domain manager, the Dynamic Workload Console, and the z/OS connector	7
Upgrading the master domain manager and the z/OS connector	7
Upgrading a master domain manager instance by performing a parallel upgrade	8

Upgrading agents	13
----------------------------	----

Chapter 3. Upgrading the Dynamic Workload Console 21

Upgrading from V8.5.1	21
Upgrading Dynamic Workload Console V8.5.1 single instance	21
Upgrading Dynamic Workload Console V8.5.1 single instance on the same workstation	21
Upgrading Dynamic Workload Console V8.5.1 single instance on a new workstation	24
Upgrading Dynamic Workload Console installed with one or more components in the same directory	27

Chapter 4. Upgrading agents in a High Availability Cluster Environment. 29

Upgrading agents from V8.5.1 in a Windows cluster environment	29
Upgrading cluster nodes automatically by using twsClusterUpg	29
Upgrading cluster nodes manually	30

Chapter 1. Steps to upgrade from V8.5.1 to V9.3

To upgrade IBM Workload Scheduler from V8.5.1 to V9.3, complete the following steps:

1. Verify that your network has the minimum required supported versions of the operating system, product, and database, as described in *IBM Workload Scheduler: Planning and Installation*, Before upgrading.
2. From the IBM Passport Advantage Online website, download the IBM Workload Scheduler V9.3 installation package.
3. From IBM Fix Central, at link: https://www-947.ibm.com/support/entry/myportal/search_results?sn=spe&filter=keywords%3Aibmsupportfixcentralsearch&q=9.3.0-TIV-TWS-UPGRADE-FP0000, download the **9.3.0-TIV-TWS-UPGRADE-FP0000** package. The package contains the following iFixes and related readme files:
 - iFix for DWC upgrade from version 8.5.1 to 9.3
 - iFix for database upgrade from version 8.5.1 to 9.3
 - iFix for direct agent upgrade from version 8.5.1 to 9.3.
4. Follow the instructions included in the iFixes readme files to add or replace some of the installation package files.
5. Proceed with the upgrade of the IBM Workload Scheduler components by following the instructions in the following sections.

Compatibility

- Master domain manager version 9.3 supports both dynamic agents and fault-tolerant agents version 8.5.1.
- Master domain manager version 8.5.1 supports fault-tolerant agents version 9.3, but not dynamic agents version 9.3.

Limitations

- For the master domain manager the direct upgrade is not supported. Only parallel upgrade from version 8.5.1 to 9.3 is supported.

Chapter 2. Upgrading IBM Workload Scheduler

Upgrading from V8.5.1

The procedure to run in the upgrade deployment model depends on whether your instance is **single** or **multiple**. A single instance contains one IBM Workload Scheduler component that is installed in the directory <TWS_INST_DIR>. A multiple instance contains two or more IBM Workload Scheduler components that are installed in the same directory <TWS_INST_DIR>.

Single instance

To upgrade a single component instance installed in the directory <TWS-INST-DIR>, you must follow the procedure listed in Table 1.

Table 1. Upgrade deployment model for IBM Workload Scheduler v8.5.1 component instances

IBM Workload Scheduler component	Procedure to run
Fault-tolerant agent V8.5.1 and related fix packs	"Upgrading agents" on page 13.
Dynamic agent V8.5.1 and related fix packs	"Upgrading agents" on page 13.
Master domain manager or its backup V8.5.1 and related fix packs	"Upgrading a master domain manager instance by performing a parallel upgrade" on page 8.
Remote command-line client V8.5.1 and related fix packs	You cannot upgrade the remote command-line client to version 9.3 directly. Perform the following procedure: <ol style="list-style-type: none">1. Save the remote command line localopts properties file.2. Install a fault-tolerant agent V9.3 instance that contains a remote command line, by performing the procedure described in Installing Agent.3. Run Configuring a remote command-line client to configure the remote command line with the old configuration properties stored in localopts.

Multiple instance

To upgrade a IBM Workload Scheduler V8.5.1 multiple components instance installed in the same directory <TWS_INST_DIR>, you must follow the procedure listed in Table 2 on page 4.

Table 2. Upgrade deployment model for IBM Workload Scheduler multiple components instances

IBM Workload Scheduler multiple components instance installed in the directory <TWS_INST_DIR>	Procedure to run
Fault-tolerant agent V8.5.1 and related fix packs Distributed Connector V8.5.1 and related fix packs	“Upgrading the fault-tolerant agent and the distributed connector” on page 5.
Fault-tolerant agent V8.5.1 and related fix packs Dynamic Workload Console V8.5.1 and related fix packs	“Upgrading the fault-tolerant agent and the Dynamic Workload Console” on page 5.
Fault-tolerant agent V8.5.1 and related fix packs Dynamic Workload Console V8.5.1 and related fix packs z/OS connector V8.5.0 or V8.5.1 and related fix packs	“Upgrading the fault-tolerant agent, the Dynamic Workload Console, and the z/OS connector” on page 5.
Fault-tolerant agent V8.5.1 and related fix packs z/OS connector V8.5.1 and related fix packs	“Upgrading the fault-tolerant agent and the z/OS connector” on page 6.
Master domain manager V8.5.1 and related fix packs Dynamic Workload Console V8.5.1 and related fix packs	“Upgrading the master domain manager and the Dynamic Workload Console” on page 6
Master domain manager V8.5.1 and related fix packs Dynamic Workload Console V8.5.1 and related fix packs z/OS connector V8.5.1 and related fix packs	“Upgrading the master domain manager, the Dynamic Workload Console, and the z/OS connector” on page 7
Master domain manager V8.5.1 and related fix packs z/OS connector V8.5.1 and related fix packs	“Upgrading the master domain manager and the z/OS connector” on page 7

Note: Due to the support of the UPN Windows user, if you have Windows domain users that are defined in the logon fields as domain\username, after performing an upgrade to version 9.3, update the Security file before starting the IBM Workload Scheduler instance. Insert the escape character '\ ' before the '\ ' character in the domain\username value.

Upgrading the fault-tolerant agent and the distributed connector

If you have a multiple components instance that contains a fault-tolerant agent and a distributed connector installed in the directory <TWS_INST_DIR>, you must first uninstall the old distributed connector and then upgrade the fault-tolerant agent in the <TWS_INST_DIR> directory.

To upgrade the multiple components instance in the correct order, perform the following steps:

1. Manually uninstall the distributed connector in the directory <TWS_INST_DIR> by using the silent or wizard uninstallation process that is provided by earlier versions of IBM Workload Scheduler.
2. Upgrade the fault-tolerant agent by using the **twinst** script as described in “Upgrading agents” on page 13.

Upgrading the fault-tolerant agent and the Dynamic Workload Console

If you have a multiple components instance that contains a Dynamic Workload Console and a fault-tolerant agent installed in the directory <TWS_INST_DIR>, you must first uninstall the old Dynamic Workload Console installed in the <TWS_INST_DIR> directory, and then upgrade the fault-tolerant agent in the <TWS_INST_DIR> directory.

To upgrade the multiple components instance in the correct order, perform the following steps:

1. Manually uninstall the old Dynamic Workload Console in the directory <TWS_INST_DIR>, by using the Dynamic Workload Console previous version uninstallation process.
2. Upgrade the fault-tolerant agent by using the **twinst** script, as described in “Upgrading agents” on page 13.

Upgrading the fault-tolerant agent, the Dynamic Workload Console, and the z/OS connector

If you installed a multiple components instance that contains a Dynamic Workload Console, a z/OS connector, and a fault-tolerant agent in the directory <TWS_INST_DIR>, you must first uninstall the Dynamic Workload Console in the old <TWS_INST_DIR> directory and then upgrade the fault-tolerant agent in the <TWS_INST_DIR> directory.

To upgrade the multiple components instance in the correct order, perform the following steps:

1. Manually uninstall the old Dynamic Workload Console in the directory <TWS_INST_DIR>, by using the Dynamic Workload Console previous version uninstallation process.
2. Manually uninstall the old z/OS connector instance in the directory <TWS_INST_DIR>, by using the z/OS connector previous version uninstallation process.
3. Upgrade the fault-tolerant agent by using the **twinst** script as described in “Upgrading agents” on page 13.

Upgrading the fault-tolerant agent and the z/OS connector

If you have a multiple components instance that contains a z/OS connector and a fault-tolerant agent in the directory <TWS_INST_DIR>, you must first configure the old z/OS connector in the <DWC_NEW_INST_DIR> directory in which you installed a new instance of the Dynamic Workload Console, uninstall the old z/OS connector and then you can upgrade the fault-tolerant agent in the <TWS_INST_DIR> directory.

To upgrade the multiple components instance in the correct order, perform the following steps:

1. Install a Dynamic Workload Console instance in the new directory <DWC_NEW_INST_DIR>, as described in Installing .
2. Export the z/OS connector configuration properties in the old multiple components instance, by running:

On Windows operating systems

```
<TWS_INST_DIR>\wastools\displayZosEngine.bat
```

On UNIX and Linux operating systems

```
<TWS_INST_DIR>/wastools/displayZosEngine.sh
```

Note: If you are connected to multiple controllers, repeat this step for each connection that you want to maintain in the new z/OS connector configuration.

3. Import the z/OS connector configuration properties into the Dynamic Workload Console instance, by running:

On Windows operating systems

```
<TWS_INST_DIR>\wastools\createZosEngine.bat
```

On UNIX and Linux operating systems

```
<TWS_INST_DIR>/wastools/createZosEngine.sh
```

Note: If you want to maintain the connections to multiple controllers, repeat this step for each connection previously defined that you want to save.

4. Manually uninstall the old z/OS connector in the directory <TWS_INST_DIR>, by using the uninstallation process provided by the earlier versions.
5. Upgrade the fault-tolerant agent by using the **twsinst** script as described in “Upgrading agents” on page 13.

Upgrading the master domain manager and the Dynamic Workload Console

You cannot make a direct upgrade of a version 8.5.1 master domain manager to a version 9.3 master domain manager. For the master domain manager, only parallel upgrade from version 8.5.1 to 9.3 is supported.

If you have a multiple components instance that contains a Dynamic Workload Console and a master domain manager installed in the directory <TWS_INST_DIR>, you must first uninstall the old Dynamic Workload Console installed in the <TWS_INST_DIR> directory, and then upgrade (with a parallel upgrade) the master domain manager in the <TWS_INST_DIR> directory.

To upgrade the shared instance in the correct order, perform the following steps:

1. Manually uninstall the old Dynamic Workload Console in the directory <TWS_INST_DIR>, by using the Dynamic Workload Console previous version uninstallation process.

2. Upgrade the master domain manager, as described in “Upgrading a master domain manager instance by performing a parallel upgrade” on page 8.

Upgrading the master domain manager, the Dynamic Workload Console, and the z/OS connector

You cannot make a direct upgrade of a version 8.5.1 master domain manager to a version 9.3 master domain manager. For the master domain manager, only parallel upgrade from version 8.5.1 to 9.3 is supported.

If you installed a multiple components instance that contains a Dynamic Workload Console, a z/OS connector, and the master domain manager in the directory <TWS_INST_DIR>, you must first uninstall the Dynamic Workload Console in the old <TWS_INST_DIR> directory, and then upgrade (with a parallel upgrade) the master domain manager in the <TWS_INST_DIR> directory.

To upgrade the multiple components instance in the correct order, perform the following steps:

1. Manually uninstall the old Dynamic Workload Console in the directory <TWS_INST_DIR>, by using the Dynamic Workload Console previous version uninstallation process.
2. Manually uninstall the old z/OS connector instance in the directory <TWS_INST_DIR>, by using the z/OS connector previous version uninstallation process.
3. Upgrade the master domain manager, as described in “Upgrading a master domain manager instance by performing a parallel upgrade” on page 8.

Upgrading the master domain manager and the z/OS connector

You cannot make a direct upgrade of a version 8.5.1 master domain manager to a version 9.3 master domain manager. For the master domain manager, only parallel upgrade from version 8.5.1 to 9.3 is supported.

If you have a multiple components instance that contains a z/OS connector and a dynamic domain manager installed in the directory <TWS_INST_DIR>, you must first configure the old z/OS connector in the <DWC_NEW_INST_DIR> directory in which you installed a new instance of the Dynamic Workload Console, uninstall the old z/OS connector, and then upgrade (with a parallel upgrade) the master domain manager in the <TWS_INST_DIR> directory.

To upgrade the multiple components instance in the correct order, perform the following steps:

1. Install a Dynamic Workload Console instance in the new directory <DWC_NEW_INST_DIR>, as described in Installing.
2. Export the z/OS connector configuration properties in the old shared instance, by running:

On Windows operating systems

```
<TWS_INST_DIR>\wastools\displayZosEngine.bat
```

On UNIX and Linux operating systems

```
<TWS_INST_DIR>/wastools/displayZosEngine.sh
```

Note: If you are connected to multiple controllers, repeat this step for each connection that you want to maintain in the new z/OS connector configuration.

3. Import the z/OS connector configuration properties in the Dynamic Workload Console instance, by running:

On Windows operating systems

```
<TWS_INST_DIR>\wastools\createZosEngine.bat
```

On UNIX and Linux operating systems

```
<TWS_INST_DIR>/wastools/createZosEngine.sh
```

Note: If you want to maintain the connections to multiple controllers, repeat this step for each connection previously defined that you want to save.

4. Manually uninstall the old z/OS connector in the directory <TWS_INST_DIR>, by using the uninstallation process provided by the earlier versions.
5. Upgrade the master domain manager, as described in “Upgrading a master domain manager instance by performing a parallel upgrade.”

Upgrading a master domain manager instance by performing a parallel upgrade

You cannot make a direct upgrade of a version 8.5.1 master domain manager to a version 9.3 master domain manager. For the master domain manager, only parallel upgrade from version 8.5.1 to 9.3 is supported.

To perform a **parallel upgrade** of the master domain manager, follow the instructions in the following sections.

Installing a new master domain manager configured as backup

Complete this procedure to install a fresh master domain manager configured as backup and link it to your current network.

The master domain manager configured as backup points to your existing IBM Workload Scheduler database and becomes your new master domain manager.

Perform the following steps:

1. **Install a master domain manager configured as backup:** For more information about how to install a new master domain manager configured as backup, see Installing a master domain manager or its backup and subsequent sections depending on whether you are using a DB2® or an Oracle database. Ensure that your new master domain manager configured as backup points to your current IBM Workload Scheduler database instance.
2. **Migrate your authentication configuration:** complete the following steps to migrate your authentication mechanism to the newly-installed master domain manager configured as backup:
 - a. On your existing master domain manager, use the `showSecurityProperties` tool to export your authentication configuration to a text file.
 - b. Copy this file to your new master domain manager configured as backup.
 - c. During the export all the passwords in the file have been replaced with asterisks. Locate them and remove the asterisks by entering passwords again.
 - d. Run the `changeSecurityProperties` tool on the new master domain manager configured as backup 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 are issued and you must configure the authentication mechanism manually.

- e. Test that the migrated authentication mechanism allows you to log on and use **composer** with more than one user ID.
3. **Define a new master domain manager configured as backup in the database:** define your new master domain manager configured as backup as a full status agent in the domain of your master domain manager, using the **composer** command interface.
4. **Prepare the old security file for switching the manager:** to switch correctly, you must add the new *IWS_user* into the old security file. The new *IWS_user* is the one that you used when you installed the new master domain manager configured as backup manager. Perform the following steps:
 - a. On the master domain manager, log in as the master *IWS_user* and set the IBM Workload Scheduler environment. Add the master configured as backup *IWS_user* to the old security file.
 - b. If you have centralized security, distribute the security file to all agents. If you do not have centralized security, copy the compiled security file *Security* to the installed master domain manager configured as backup, overwriting the version that is there.
5. **Distribute the Symphony file to the new master domain manager configured as backup:**
 - a. Ensure that the **optman cf** option is set to *all*.
 - b. To distribute the Symphony file to the new master domain manager configured as backup, run **JnextPlan -for 0000** or wait until the end of the production plan.
 - c. Restore the previous setting of the **optman cf** option, if you previously modified the value.

Switching the master domain manager to the new backup master

Switch to your new backup master domain manager, which now becomes your master domain manager, by issuing the following command from the old master domain manager:

```
conman  
switchmgr masterdm;new_mgr_cpu
```

where *new_mgr_cpu* is the backup master domain manager workstation name.

From the Dynamic Workload Console, in the navigation tree, click **System Status and Health > Monitor Workload** > select the engine and the object type Workstation, click run and, in the table of results, select backup master domain manager workstation name, click **More Actions**, and select **Become Master Domain Manager**.

Switch the event processor from the old master domain manager to the backup master domain manager, by running the following command from the old master domain manager:

```
conman  
switcheventprocessor new_mgr_cpu
```

where *new_mgr_cpu* is the backup master domain manager workstation name.

From the Dynamic Workload Console, in the navigation tree, click **System Status and Health > Monitor Workload** > select the engine and the object type

Workstation, click run and, in the table of results, select backup master domain manager workstation name, click **More Actions**, and select **Become Event Processor**.

Note: The new backup master domain manager connecting to the old database updated schemas and tables to ensure product capability with earlier versions. Therefore, the old master domain manager and the new backup master domain manager work with the new updated schema.

Making the switch manager permanent

In the procedure “Switching the master domain manager to the new backup master” on page 9, you switched your master domain manager promoting your new version backup master domain manager to the role of master domain manager.

To make this configuration fully operational and persistent through **JnextPlan**, you must 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 domain manager. For more information about *localopts* file, see the *IBM Workload Scheduler: Administration Guide*.

2. Change the workstation definition of the old master by running:

```
composer modify cpu=old_mgr_cpu
```

and in the definition substitute *type=manager* with *type=fta*

3. Change the workstation definition of the new master by running:

```
composer modify cpu=new_mgr_cpu
```

and in the definition substitute *type=fta* with *type=manager*.

4. Ensure that the **optman** cf option is set to *all*.
5. Rebuild the plan to activate the changes to the database:

```
JnextPlan -for 0000
```
6. Restore the previous setting of the **optman** cf option, if necessary.
7. 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 *IBM Workload Scheduler: Administration Guide*.

In this way the reports **reptr-pre** and **reptr-post** can run when you run **JnextPlan**.

8. Ensure that the global option **carryforward** is set to **all** or only the unfinished jobstreams are carried forward.

Customizing and submitting the optional final job stream

Some steps that should be followed when customizing and submitting the optional final job stream.

The upgrade process writes the latest FINAL and FINALPOSTREPORTS definitions for the current release in the following file: <TWA_HOME>/TWS/config/Sfinal, where <TWA_HOME> is the IBM Workload Scheduler installation directory. To use these latest definitions, you must merge the functions of your current FINAL and FINALPOSTREPORTS job streams with the syntax of your new FINAL and FINALPOSTREPORTS job streams. Perform the following steps:

1. Customize the final job stream as required:

If you had customized job streams called FINAL and FINALPOSTREPORTS in your database:

- a. Extract the definitions from the current FINAL and FINALPOSTREPORTS job streams file using **composer**.
- b. Use a text editor to edit your customized FINAL and FINALPOSTREPORTS job streams.
- c. Merge the job streams with file <TWA_HOME>/TWS/config/Sfinal so that the new FINAL and FINALPOSTREPORTS job streams have the same customization as your customized final job streams plus the new required attributes provided by the new FINAL and FINALPOSTREPORTS job streams.
- d. Save your new FINAL and FINALPOSTREPORTS job streams using **composer**.

If you had customized final job streams called something other than FINAL and FINALPOSTREPORTS in your database:

- a. Extract the definitions from your customized final job stream files using **composer**.
- b. Use a text editor to edit your customized final job stream files.
- c. Merge the job streams with file <TWA_HOME>/TWS/config/Sfinal so that the new FINAL and FINALPOSTREPORTS job streams have the same customization as your customized final job streams plus the new required attributes provided by the new FINAL and FINALPOSTREPORTS job streams.
- d. Save these new final job streams so that they have the same names as your current customized final job streams by running the command **composer -replace**.

If you had final job streams called something other than FINAL and FINALPOSTREPORTS in your database, but they are not customized:

- a. Make a copy of file <TWA_HOME>/TWS/config/Sfinal.
- b. Edit this copy and rename the FINAL and FINALPOSTREPORTS parameters with the actual names .
- c. Run the command **composer -replace**.

If you had final job streams called FINAL and FINALPOSTREPORTS in your database, but they are not customized:

Run the command **composer -replace <TWA_HOME>/TWS/config/Sfinal**.

If you had final job streams called FINAL and FINALPOSTREPORTS but they are in DRAFT in your database:

Run the command **composer -replace** and, after the upgrade, change these job streams into the DRAFT status again.

2. Use **conman** to delete your current 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

Upgrading your old master domain manager

After you switched the master domain manager to the new backup master, either maintain your old master domain manager at the original version or perform a direct upgrade to version 9.1.

Completing the security configuration for the new environment

If you have specific security settings in your current 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.

To manually merge the new settings, perform the following procedure:

1. Log in as `<TWS_user>` on your upgraded master domain manager and set the IBM Workload Scheduler environment.
2. If you have centralized security enabled, extract the new security file on the new master using the 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:

Workload application

```
WKLDAPPL NAME=@ ACCESS=ADD,DELETE,DISPLAY,MODIFY,LIST,UNLOCK
```

Run cycle group

```
RUNCYGRP NAME=@ ACCESS=ADD,DELETE,DISPLAY,MODIFY,USE,LIST,UNLOCK
```

Centralized agent update

Replace the statement:

```
CPU CPU=@ ACCESS=ADD,CONSOLE,DELETE,DISPLAY,FENCE,LIMIT,LINK,MODIFY,SHUTDOWN,  
START,STOP,UNLINK,LIST,UNLOCK,RUN,RESETFTA
```

with the following statement:

```
CPU CPU=@ ACCESS=ADD,CONSOLE,DELETE,DISPLAY,FENCE,LIMIT,LINK,MODIFY,SHUTDOWN,  
START,STOP,UNLINK,LIST,UNLOCK,RUN,RESETFTA,MANAGE
```

4. Check that the user permissions of the new statements are correct.
5. Due to new support of the UPN Windows user, if you have Windows domain users that are defined in the logon fields as `domain\username`, insert the escape character `'\'` before the `'\'` character in the `domain\username` value. For example, if you use the `MYDOMAIN\user1` value in the logon field, after the upgrade, in the Security file you must update the line in following way:

```
.....  
logon=MYDOMAIN\\user1  
.....
```

6. Save your changes to the `sec_file`.
7. Build your final security file for your new master domain manager using the **makesec** command:

```
makesec sec_file
```

8. If you are using FIPS, you must manually enable it again in the WebSphere Application Server `java.security` file. For the FIPS compliance information, see the *IBM Workload Scheduler: Administration Guide* .
9. If you have centralized security enabled, distribute the security file.
10. 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 are carried forward.

11. Restore the previous setting of the `optman cf` option, if necessary.

Upgrading agents

Note: Master domain manager version 9.3 supports both dynamic agents and fault-tolerant agents version 8.5.1. Master domain manager version 8.5.1 supports fault-tolerant agents version 9.3, but not dynamic agents version 9.3.

The product performs the upgrade in safe mode by performing all the checks detailed in Performing a safe upgrade before starting.

To upgrade agents perform the following steps:

1. “Unlinking and stopping IBM Workload Scheduler when upgrading agent workstations”
2. “Running the upgrade” on page 14.

Note: During the upgrade, you can add dynamic scheduling capabilities or the Java™ run time to run job types with advanced options to the agent. The run time 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 run remotely, from the agent, the dynamic workload broker resource command on the server.

The upgrade process changes some files and folders. For the complete list, see Files and folders changed during the upgrade .

Note: When the upgrade procedure has completed successfully, it is not possible to roll back to the previous version.

Unlinking and stopping IBM 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 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 IBM 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 complete.

Note: Do not use the UNIX `kill` command to stop IBM Workload Scheduler processes.

To stop IBM Workload Scheduler processes and services, follow these steps:

1. Unlink the target workstation by entering the following command from the command line of the master domain manager:
`conman "unlink workstationname"`
2. To stop the target workstation, from the command line of the master domain manager log in as *TWS_user* and enter the following command:
`conman "stop workstationname;wait"`
3. Stop the **netman** process as follows:
 - On Windows operating systems, from the IBM Workload Scheduler home directory, run the command:
`shutdown.cmd`
 - On UNIX operating systems, run the command:
`conman "shut;wait"`
4. If you are updating an agent, remove (unmount) any NTFS mounted directories from the master domain manager.

To check if any services and processes are still running:

- On Windows operating systems, enter the command:

```
<IWA_HOME>\unsupported\listproc.exe
```

Verify that the following processes are not running: netman, mailman, batchman, writer, jobman, stageman, JOBMON, tokensrv, batchup, monman.

All processes must be stopped.

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 operating systems, enter the command:
`ps -u IWS_user`

Running the upgrade

The upgrade process meets the following objectives:

Performs the upgrade in a safe way

It checks for any 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. For detailed information, see Performing a safe upgrade.

Saves 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.

Uses a very simple command

It consists of a single line command.

Manages both Windows and UNIX operating system workstations

It runs on both Windows and UNIX agents.

Use the **twinsinst** script to upgrade the IBM Workload Scheduler agent in your distributed or end-to-end network and add dynamic scheduling capabilities or the Java run time to run job types with advanced options to the agent. The run time 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 run remotely, from the agent, the dynamic workload broker resource command on the server.

To add dynamic scheduling capabilities, specify the **tdwbport** and **tdwbhostname** parameters as described in Agent installation parameters . To add the Java run time to run job types with advanced options to the agent, specify the **-addjruntime** parameter as described in Agent installation parameters .

For information about agents installed using the **twinsinst** script, see Installing agents .

For a list of supported operating systems and requirements, see the System Requirements Document at <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27045181> .

Procedure:

1. Before starting to install, upgrade, or uninstall, verify that the user running the installation process has the following authorization requirements:

Windows operating systems

If you set the Windows User Account Control (UAC), your login account must be a member of the Windows **Administrators** group or domain administrators with the rights **Act as Part of the Operating System**.

If you set the Windows User Account Control (UAC) on the workstation, you must run the installation as **administrator**.

UNIX and Linux operating systems

You must have **root** access.

2. Ensure that you inserted the DVD for your operating system or that you downloaded the IBM Workload Scheduler agent eImage (for details, see the Download Document at <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg24039466>).
3. Ensure that you have enough temporary space before starting the installation process. If you have not much space in the temporary directory and you cannot free the space, see **twinsinst** needs long time to run if the machine does not have enough temporary space .

To upgrade agents using the **twinsinst** script, perform the following steps:

On Windows operating systems

1. Insert the DVD for your operating system. See Installation media .
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 the **twinsinst** using the synopsis described below.

Note: **twinsinst** for Windows is a Visual Basic Script (VBS) that you can run in CScript and WScript mode, for example:

```
cscript twsinst -update -uname username -password password -acceptlicense yes
```

If you enabled the Security Warning, a dialog box is displayed during the upgrade. In this case answer **Run** to continue.

On UNIX and Linux operating systems

1. Insert the installation DVD for your operating system. See Installation media.
2. From the *DVD_root/TWS/operating_system* directory, run the **twsinst** script as described in Synopsis.

A successful upgrade using the **twsinst** script issues the return code RC = 0. If the upgrade fails, to understand the cause of the error see Synopsis.

Synopsis:

On Windows operating systems

Show command usage and version

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

Upgrade an instance

```
twsinst -update -uname user_name  
-password user_password  
-acceptlicense yes|no  
[-addjruntime true]  
[-displayname agentname]  
[-hostname host_name]  
[-inst_dir install_dir [-recovInstReg true]]  
[-jimport port_number]  
[-jimportssl boolean]  
[-lang lang_id]  
[-skipcheckprereq]  
[-skip_usercheck]  
[-tdwbhostname host_name]  
[-tdwbport port_number]  
[-wait minutes]  
[-work_dir working_dir]
```

On UNIX and Linux operating systems

Show command usage and version

```
./twsinst -u | -v
```

Upgrade an instance

```
./twsinst -update [-uname user_name]  
-acceptlicense yes|no  
[-addjruntime true]  
[-create_link]  
[-displayname agentname]  
[-hostname host_name]  
[-inst_dir install_dir [-recovInstReg true]]  
[-jimport port_number]  
[-jimportssl boolean]  
[-lang lang-id]  
[-reset_perm]  
[-skipcheckprereq]  
[-skip_usercheck]  
[-tdwbhostname host_name]  
[-tdwbport port_number]  
[-wait minutes]  
[-work_dir working_dir]
```

-acceptlicense *yes|no*

Specify whether or not to accept the License Agreement.

-addjruntime *true*

Adds the Java run time to run job types with advanced options to the agent. The run time environment is used to run application job plug-ins on the agent and to enable the capability to run remotely, from the agent, the dynamic workload broker resource command on the server.

This option is applicable to both fault-tolerant agents and dynamic agents.

By default, if the Java run time was already installed on the agent, it is upgraded to version 9.3.

If the Java run time was not installed on the agent, it is not installed during the upgrade, unless you specify `-addjruntime true`.

-create_link

UNIX operating systems only. Create the **symlink** between `/usr/bin/at` and `<install_dir>/TWS/bin/at`.

-displayname

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

-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 IBM Workload Scheduler. When upgrading, the directory **inst_dir** is used whether:

- The upgrade process cannot retrieve the product install location from the registries.
- You need to create the IBM Workload Scheduler registries again before upgrading. See Re-creating registry files using `twsinst` for details.

If you do not provide the **inst_dir** directory and IBM Workload Scheduler cannot retrieve it from the installation registries, the product is installed in the user home directory.

On Windows operating systems:

If you specify a path that contains blanks, enclose it in double quotation marks. If not specified, the path is set to `%ProgramFiles%\IBM\TWA`.

On UNIX and Linux operating systems:

The path cannot contain blanks. If not specified, the path is set to the *user_name* home directory.

-jimport

The port used by the IBM Workload Scheduler for z/OS® server or the dynamic workload broker to connect to the IBM Workload Scheduler agent. The default value is **31114**. The valid range is from 1 to 65535.

-jimportssl

The port used by the IBM Workload Scheduler for z/OS controller or by the dynamic workload broker to connect to the IBM Workload Scheduler agent. This number is registered in the `ita.ini` file located in the directory `ITA\cpa\ita` on Windows systems and the directory `ITA/cpa/ita` on UNIX systems. For communication using SSL, set `jimportssl` to *true*. To communicate with the dynamic workload broker, it is recommended that you set the value to **true**. In this case, the port specified in **jimport** communicates in HTTPS. 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 **jimport** communicates in HTTP. The default value is **true**. For communication without using SSL, set *jimportssl* to *false*. To increase the performance of the IBM Workload Scheduler for z/OS server, it is recommended that you set this value to false.

-lang

The language in which the **twinst** messages are displayed. If not specified, the system LANG is used. If the related catalog is missing, the default C language catalog is used.

Note: The **-lang** option does not relate to the supported language packs. By default, all supported language packs are installed when you install using the **twinst** script.

-password

Windows system only. The password of the user for which you are installing IBM Workload Scheduler.

-recovInstReg true

To re-create the registry files. Specify if you tried to upgrade a stand-alone, fault-tolerant agent (an agent that is not shared with other components or does not have the connector feature) and you received an error message that states that an instance of IBM Workload Scheduler cannot be found. This error can be caused by a corrupt registry file. See *Upgrading when there are corrupt registry files* . If you specify this parameter you must set **-inst_dir** option.

-reset_perm

UNIX systems only. Reset the permissions of the **libatrc** library.

-skipcheckprereq

If you specify this parameter, IBM Workload Scheduler does not scan system prerequisites before installing the agent. For more information on the prerequisite check, see *Scanning system prerequisites for IBM Workload Scheduler* .

-skip_usercheck

Enable this option if the authentication process within your organization is not standard, thereby disabling the default authentication option. On UNIX and Linux operating systems if you specify this parameter, the program skips the check of the user in the */etc/passwd* file or the check you perform using the **su** command. On Windows operating systems if you specify this parameter, the program does not create the user you specified in the **-uname username** parameter. If you specify this parameter you must create the user manually before running the script.

-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 IBM 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 from 0 to 65535.

-uname *username*

The name of the user for which IBM 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 the **twsinst** script.

-wait *minutes*

The number of minutes that the product waits for jobs that are running to complete before starting the upgrade. If the jobs do not complete during this interval the upgrade does not proceed and an error message is displayed. Valid values are integers or **-1** for the product to wait indefinitely. The default is **60**.

-work_dir *working_dir*

The temporary directory used for the IBM Workload Scheduler upgrade process files deployment.

On Windows operating systems:

If you specify a path that contains blanks, enclose it in double quotation marks. If you do not manually specify a path, the path is set to %temp%\TWA\tws93, where %temp% is the temporary directory of the operating system.

On UNIX and Linux operating systems:

The path cannot contain blanks. If you do not manually specify a path, the path is set to /tmp/TWA/tws93.

Chapter 3. Upgrading the Dynamic Workload Console

Upgrading from V8.5.1

This section provides an overview of the upgrade process of an existing version of Dynamic Workload Console.

Upgrade deploy model for single or multiple V8.5.1 component instances installed in the directory <TWS_INST_DIR>:

Single instance:

A single instance contains the Dynamic Workload Console component installed in the <TWS_INST_DIR> directory.

Multiple instance:

A multiple Instance contains the Dynamic Workload Console component and one or more IBM Workload Scheduler components installed in the same <TWS_INST_DIR> directory.

You must know if the instance you are upgrading is *single* or *multiple* to understand which procedure you must use to upgrade the Dynamic Workload Console.

Upgrading Dynamic Workload Console V8.5.1 single instance

You can upgrade a single instance of Dynamic Workload Console V8.5.1 in one of the following ways:

Procedure to upgrade a Dynamic Workload Console on the same workstation where the back-level is installed:

“Upgrading Dynamic Workload Console V8.5.1 single instance on the same workstation.”

Procedure to upgrade Dynamic Workload Console on a new workstation:

“Upgrading Dynamic Workload Console V8.5.1 single instance on a new workstation” on page 24

Upgrading Dynamic Workload Console V8.5.1 single instance on the same workstation

To upgrade a single instance of Dynamic Workload Console on the same workstation where the back level Dynamic Workload Console is installed, run the following steps:

1. Install a new Dynamic Workload Console in the <DWC_INSTALL_DIR> directory, on the system where the back-level Dynamic Workload Console is installed.
For information about Dynamic Workload Console installation, see Installation procedure for Dynamic Workload Console .
2. Ensure that no Dynamic Workload Console user interface is active and that the WebSphere Application Server is up and running.
3. Migrate the data from the back-level to the newly installed Dynamic Workload Console by running the following script:

On Windows operating systems:

```

<DWC_INSTALL_DIR>\TDWC\scripts\tdwcUpgrade.bat
  -oldwasuser old_user
  -oldwaspassword old_password
  -oldtwapath old_twa_path
  -newwasuser new_user
  -newwaspassword new_password
  -newtwapath new_twa_path
  [-backuppath backup_path]
  [-machinechange false]

```

On UNIX and Linux operating systems:

```

<DWC_INSTALL_DIR>/TDWC/scripts/tdwcUpgrade.sh
  -oldwasuser old_user
  -oldwaspassword old_password
  -oldtwapath old_twa_path
  -newwasuser new_user
  -newwaspassword new_password
  -newtwapath new_twa_path
  [-backuppath backup_path]
  [-machinechange false]

```

where:

- oldwasuser** *old_user*
The Tivoli Integrated Portal administrator user ID specified for the back-level Dynamic Workload Console.
- oldwaspassword** *old_password*
The Tivoli Integrated Portal administrator user password specified for the back-level Dynamic Workload Console.
- oldtwapath** *old_twa_path*
The installation directory where the back-level Dynamic Workload Console is installed.
- newwasuser** *new_user*
The Dashboard Application Services Hub administrator user ID.
- newwaspassword** *new_password*
The password of the Dashboard Application Services Hub administrator.
- newtwapath** *new_twa_path*
The installation directory where you want to install the Dynamic Workload Console. By default the installation directory is:

On Windows operating systems:

C:\Program Files\IBM\TWAUI

On UNIX and Linux operating systems:

/opt/IBM/TWAUI

- backuppath** *backup_path*
The <BACKUP_DIR> backup directory for the upgrade. By default the backup directory is:

On Windows operating systems:

<DWC_INSTALL_DIR>\TDWC\tmp\backup

On UNIX and Linux operating systems:

<DWC_INSTALL_DIR>/TDWC/tmp/backup

This directory contains:

- The **tdwcUpgrade** script log file, `upgrade.log`.

- The files containing the following configuration data exported from the back-level Dynamic Workload Console:
 - The embedded WebSphere Application Server profile in the UpgradeData.zip file.
 - The embedded WebSphere Application Server profile registry.
 - The port settings in the ports.txt file.
 - The Tivoli Integrated Portal settings.
 - The Dynamic Workload Console settings.

This data is then imported into the newly installed Dynamic Workload Console.

- A compressed file named backup.zip containing a saved copy of the configuration data of the newly-installed Dynamic Workload Console. This file is used to roll back to the original configuration if the migration script fails while importing the configuration data from the back-level Dynamic Workload Console.

Note: Because the backup.zip file is overwritten every time you run the migration script, it might be useful to save a copy of the first backup.zip file containing the original configuration.

-machinechange *false*

To upgrade on the same workstation you must specify the false value. The default value is false.

Note:

The script replaces any customized data in the new Dynamic Workload Console instance, with the data exported from the old Dynamic Workload Console instance.

A result of Completed indicates that the script ran successfully and that the data was correctly imported into the newly-installed Dynamic Workload Console.

If the script fails to import the configuration data into the newly-installed Dynamic Workload Console, a rollback is automatically performed, and the original configuration is restored. To double-check that the rollback ran correctly, ensure that you can access the newly-installed Dynamic Workload Console user interface with the user ID and password specified during the installation.

4. The port numbers used by the two instances of the Dynamic Workload Console are different and they are not automatically migrated by the **tdwcUpgrade** script. Run the following steps to migrate the port numbers of the back-level instance to the newly-installed instance:
 - a. Check that the data was correctly migrated from the old Dynamic Workload Console to the newly-installed Dynamic Workload Console.
 - b. Stop the back-level Dynamic Workload Console.
 - c. Uninstall the back-level Dynamic Workload Console.
 - d. Run the following command:

On Windows operating systems:

```
changeHostProperties <BACKUP_DIR>\ports.txt
```

On UNIX and Linux operating systems:

```
changeHostProperties <BACKUP_DIR>/ports.txt
```

where <BACKUP_DIR> is the backup directory.

Upgrading Dynamic Workload Console V8.5.1 single instance on a new workstation

To upgrade a Dynamic Workload Console installed in the directory <DWC_BACKLEV_INSTALL_DIR> on your old workstation into the directory <DWC_INSTALL_DIR> on the new workstation, run the following steps:

1. Log on as Administrator on Windows operating systems, or as root on UNIX and Linux operating systems, on the new workstation where you want to upgrade the Dynamic Workload Console.
2. Install a new Dynamic Workload Console in the <DWC_INSTALL_DIR> directory on the new workstation.

For information about Dynamic Workload Console installation, see Installation procedure for Dynamic Workload Console.

3. Ensure that no Dynamic Workload Console user interface is active and that the WebSphere Application Server is up and running on the workstation where the back-level is installed.
4. From the new workstation perform the following steps:

On Windows operating systems:

Map the network drive <DWC_BACKLEV_INSTALL_DIR> of the old workstation where the back-level Dynamic Workload Console is installed.

On UNIX and Linux operating systems:

Mount in read-write access the remote file system <DWC_BACKLEV_INSTALL_DIR> where the back-level Dynamic Workload Console is installed. If the mount point name on the new workstation is different from the remote file system name, create a symbolic link between the mount point on the new workstation and the remote file system <DWC_BACKLEV_INSTALL_DIR>, the link name value must be <DWC_BACKLEV_INSTALL_DIR>.

5. Migrate the data from the back-level to the newly-installed Dynamic Workload Console by running the following script from the new workstation:

On Windows operating systems:

```
<DWC_INSTALL_DIR>\TDWC\scripts\tdwclUpgrade.bat
-oldwasuser old_user
-oldwaspassword old_password
-oldtwapath old_twa_path
-newwasuser new_user
-newwaspassword new_password
-newtwapath new_twa_path
[-backuppath backup_path]
-machinechange true
```

On UNIX and Linux operating systems:

```
<DWC_INSTALL_DIR>/TDWC/scripts/tdwclUpgrade.sh
-oldwasuser old_user
-oldwaspassword old_password
-oldtwapath old_twa_path
-newwasuser new_user
-newwaspassword new_password
-newtwapath new_twa_path
[-backuppath backup_path]
-machinechange true
```

Where:

- oldwasuser** *old_user*
The Tivoli Integrated Portal administrator user ID specified for the back-level Dynamic Workload Console.
- oldwaspassword** *old_password*
The Tivoli Integrated Portal administrator user password that is specified for the back-level Dynamic Workload Console.
- oldtwapath** *old_twa_path*
The installation directory where the back-level Dynamic Workload Console is installed.
- newwasuser** *new_user*
The Dashboard Application Services Hub administrator user ID.
- newwaspassword** *new_password*
The password of the Dashboard Application Services Hub administrator.
- newtwapath** *new_twa_path*
The installation directory where the Dynamic Workload Console must be installed. By default the installation directory is:
 - On Windows operating systems:**
C:\Program Files\IBM\TWAUI
 - On UNIX and Linux operating systems:**
/opt/IBM/TWAUI
- backuppath** *backup_path*
The <BACKUP_DIR> backup directory for the upgrade process.
By default the installation directory is:
 - On UNIX and Linux operating systems:**
<DWC_INSTALL_DIR>/TDWC/tmp/backup
 - On Windows operating systems:**
<DWC_INSTALL_DIR>\TDWC\tmp\backup

This directory contains:

- The **tdwcUpgrade** script log file, `upgrade.log`.
- The files containing the following configuration data exported from the back-level Dynamic Workload Console:
 - The embedded WebSphere Application Server profile.
 - The embedded WebSphere Application Server profile registry.
 - The Tivoli Integrated Portal settings.
 - The Dynamic Workload Console settings.

This data is then imported into the newly installed Dynamic Workload Console.

- A compressed file named `backup.zip` containing a saved copy of the configuration data of the newly-installed Dynamic Workload Console. This file is used to roll back to the original configuration if the migration script fails while importing the configuration data from the back-level Dynamic Workload Console.

Note: Because the `backup.zip` file is overwritten every time you run the migration script, it might be useful to save a copy of the first `backup.zip` file containing the original configuration.

-machinechange *true*

You must specify the true value to upgrade the Dynamic Workload Console on the new workstation.

Note: The script replaces any customized data in the new Dynamic Workload Console instance, with the data exported from the old Dynamic Workload Console instance. A result of Completed indicates that the script ran successfully and that the data was correctly imported into the newly-installed Dynamic Workload Console.

If the script fails to import the configuration data into the newly-installed Dynamic Workload Console, a rollback is automatically performed and the original configuration is restored. To double-check that the rollback ran correctly, ensure that you can access the newly-installed Dynamic Workload Console user interface with the user ID and password specified during the installation.

6. Check that the data was correctly migrated from the old Dynamic Workload Console to the newly-installed Dynamic Workload Console.
7. Perform this step only if the Dynamic Workload Console ports of the instance installed in the new workstation are different from these of the instance installed on the old workstation and you want to have the same values.

The port numbers used by the two instances of the Dynamic Workload Console might be different and they are not automatically migrated by the **tdwcUpgrade** script. If the Dynamic Workload Console ports of the instance installed on the new workstation are different from those installed on the old machine and you want to have the same values, run the following steps to migrate the port numbers of the back level instance on the old machine to the newly-installed instance on a new machine:

- a. Log on as Administrator on Windows operating systems, or as root on UNIX and Linux operating systems, on the workstation where the back-level Dynamic Workload Console is installed.
- b. To save the Dynamic Workload Console ports data, redirect the **showHostProperties** script output to the `HostProperties_file` file:

On Windows operating systems:

From the `<DWC_BACKLEV_INSTALL_DIR>\wastools:`

```
showHostProperties.sh  
--username old_DWCuser  
--password old_DWCpassword
```

> `HostProperties_file`

On UNIX and Linux operating systems:

From the `<DWC_BACKLEV_INSTALL_DIR>/wastools:`

```
showHostProperties.bat  
--username old_DWCuser  
--password old_DWCpassword
```

> `HostProperties_file`

where:

old_DWCuser

The Tivoli Integrated Portal administrator user ID specified for the back-level Dynamic Workload Console.

old_DWCpassword

The Tivoli Integrated Portal administrator user password specified for the back-level Dynamic Workload Console.

- c. Copy the `HostProperties_file` created in step 7b on page 26 in the old workstation to the new workstation in the `<BACKUP_DIR>` backup directory that you used for the upgrade process. By default the backup directory used in the upgrade process is:

On Windows operating systems:

`<DWC_INSTALL_DIR>\TDWC\tmp\backup`

On UNIX and Linux operating systems:

`<DWC_INSTALL_DIR>/TDWC/tmp/backup`

- d. To import the old workstation port values to the Dynamic Workload Console installed on the new workstation, run:

On Windows operating systems:

`changeHostProperties <BACKUP_DIR>\HostProperties_file`

On UNIX and Linux operating systems:

`changeHostProperties <BACKUP_DIR>/HostProperties_file`

- 8. Optionally, uninstall the back-level Dynamic Workload Console on the old machine.

Upgrading Dynamic Workload Console installed with one or more components in the same directory

If you want to upgrade Dynamic Workload Console installed with one or more components in the same directory `<TWS_INST_DIR>` with Dynamic Workload Console, you must follow the procedures listed in Table 3 that also apply when upgrading from V8.6.0.

Table 3. Upgrade deployment model for Dynamic Workload Console multiple components instance in the same directory

Dynamic Workload Console and multiple IBM Workload Scheduler components instance, installed in the <code><TWS_INST_DIR></code> directory contains:	Procedure to run:
Fault-tolerant agent V8.5.1 and related Fix Packs Dynamic Workload Console V8.5.1 and related Fix Packs	Upgrading the fault-tolerant agent and the Dynamic Workload Console .
Fault-tolerant agent V8.5.1 and related Fix Packs Dynamic Workload Console V8.5.1 and related Fix Packs z/OS connector V8.5.1 and related Fix Packs	Upgrading the fault-tolerant agent, the Dynamic Workload Console, and the z/OS connector.

Table 3. Upgrade deployment model for Dynamic Workload Console multiple components instance in the same directory (continued)

<p>Dynamic Workload Console and multiple IBM Workload Scheduler components instance, installed in the <TWS_INST_DIR> directory contains:</p>	<p>Procedure to run:</p>
<p>master domain manager V8.5.1 and related Fix Packs</p> <p>Dynamic Workload Console V8.5.1 and related Fix Packs</p>	<p>Upgrading the master domain manager and the Dynamic Workload Console.</p>
<p>master domain manager V8.5.1 and related Fix Packs</p> <p>Dynamic Workload Console V8.5.1 and related Fix Packs</p> <p>z/OS connector V8.5.1 and related Fix Packs</p>	<p>Upgrading the master domain manager, the Dynamic Workload Console, and the z/OS connector.</p>
<p>Dynamic Workload Console V8.5.1 and related Fix Packs</p> <p>z/OS connector V8.5.1 and related Fix Packs</p>	<p>Upgrading the Dynamic Workload Console and the z/OS connector.</p>

Chapter 4. Upgrading agents in a High Availability Cluster Environment

Upgrading agents from V8.5.1 in a Windows cluster environment

You can upgrade cluster nodes to the general availability (GA) version of the product either automatically, by running **twscClusterUpg** script, or manually.

Upgrading cluster nodes automatically by using **twscClusterUpg**

Use the **twscClusterUpg** command to upgrade automatically IBM Workload Scheduler agents in a cluster-aware configuration, from V8.5.1 to V9.3.

To perform the upgrade, run the following procedure:

1. Ensure that your environment meets the prerequisites that are listed in Prerequisites .
2. Move to the directory where you downloaded the images of the general availability (GA) version.
3. On all the nodes you want to upgrade, run the **twscClusterUpg** script with Cluster Administrator and Domain Administrator rights by using the following syntax:

```
cscript.exe twscClusterUpg.vbs
-groups <group1, ..., groupn>
-passwords <pwd1, ..., pwn>
```

Where:

group1, ..., groupn

Specify the groups that you want to upgrade.

pwd1, ..., pwn

Specify the password of the IBM Workload Scheduler user you had used to install the product in the group. Specify the passwords in the same order that you specify the groups.

The script pauses all the nodes where IBM Workload Scheduler is installed and proceeds to upgrade the product group by group. If errors occur, analyze the %temp%\TWA\TWSClusterMainLog.txt file.

Where %temp% is the temporary directory of the user that is running the **twscClusterUpg** command.

The **twscClusterUpg** command runs the **twscinst.vbs**, the **twscpatch.vbs**, and the **twscClusterAdm.exe** commands. If the TWSClusterMainLog.txt indicates that one of these commands fails, analyze the corresponding log for further details.

4. Resume all the nodes of the cluster you upgraded, by running the following command against each node:

```
cluster.exe node <node_name> /Resume
```

Where *node_name* is the name of the node you want to resume.

5. Bring online the IBM Workload Scheduler resources on all the nodes, by running the following command against each resource:

```
cluster.exe res <res_name> /Online
```

Where *<res_name>* is the name of the IBM Workload Scheduler resource.

Upgrading cluster nodes manually

You can upgrade IBM Workload Scheduler agents in a cluster-aware configuration, from V8.5.1 to V9.3 by running the following manual procedure.

1. Ensure your environment meets the prerequisites listed in Prerequisites .
2. Set all the nodes of the cluster to the **Pause** state. You must pause all the nodes in which you defined at least one cluster resource. In this way the IBM Workload Scheduler cluster resources cannot be moved from one node to another. With this operation you fix the resources on the node where they are running. Perform this action by running the following command against each node:

```
cluster.exe node <node_name> /Pause
```

Where *<node_name>* is the name of the node to pause.

3. Set offline all the IBM Workload Scheduler resources belonging to the nodes of the cluster you paused to prevent IBM Workload Scheduler from upgrading the resources DLL with a cluster service that is still loading the DLL. Perform this action by running the following command against each resource:

```
cluster.exe res <res_name> /Offline
```

Where *<res_name>* is the name of the resource to set offline.

4. Move to the directory where you downloaded the images of the general availability (GA) version.
5. For all the nodes in the cluster, upgrade all the resources running on the nodes, by performing the following steps:
 - a. Generate the installation registries for one of the cluster groups belonging to the node on which you are upgrading the resources, and upgrade the instance you are working on, using one of the following script:

```
cscript.exe twsinst.vbs -update -uname <user_name>  
-password <TWS_user_password>  
-acceptlicense yes  
-inst_dir TWS_install_dir -recovInstReg true
```

- b. Move to the *inst_dir* directory where you upgraded the cluster node and update the remote Windows Services and the resource DLL, by running the following command:

```
twsClusterAdm.exe -update resource=<res_name> ask=yes  
-twsupd pwd <TWS_user_password>
```

Where *<res_name>* is the name of the resource you are upgrading and *<TWS_user_password>* is the Windows password of the IBM Workload Scheduler user.

Note: If you added a node to your IBM Workload Scheduler cluster after installing IBM Workload Scheduler, run the following command:

```
twsClusterAdm.exe -update hosts=<hostname1>,<hostname2>,...  
resource=<res_name> ask=yes -twsupd pwd <TWS_user_password>
```

Where *<hostname1>*,*<hostname2>*,... are the host names of the cluster nodes that you added after the installation and you want to upgrade.

- c. Repeat steps 5a and 5b for all the resources present on this node.
6. Resume all the nodes of the cluster you paused in the Step 1 by running the following command against each node:

```
cluster.exe node <node_name> /Resume
```

Where <node_name> is the name of the node you want to resume.

7. Bring online the IBM Workload Scheduler resources on all the nodes, by running the following command against each resource:

```
cluster.exe res <res_name> /Online
```

Where <res_name> is the name of the IBM Workload Scheduler resource.



Printed in USA