

IBM WebSphere Application Server



Readme for the update installer application for Version 5.0.2

Note

Before using this information, be sure to read the general information under Chapter 6, "Notices", on page 63.

Compilation date: June 19, 2003

© Copyright International Business Machines Corporation 2003. All rights reserved.

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

Contents

About this readme_updateinstaller document	v
Chapter 1. Installing interim fixes and fix packs	1
Installing interim fixes and fix packs on the base product	1
Installing interim fixes and fix packs on the Network Deployment product	4
Installing interim fixes and fix packs on the Enterprise product	7
Chapter 2. Uninstalling fixes and fix packs	13
Removing fixes and fix packs from the base product	13
Removing fixes and fix packs from the Network Deployment product	15
Removing fixes and fix packs from the Enterprise product	17
Chapter 3. UpdateSilent command	21
Syntax examples	26
Help	26
Use a properties file to supply values	26
Fix processing	26
View applied fixes	26
View available fixes.	27
Fix pack processing	27
View applied fix packs.	27
View available fix packs	27
Parameters.	27
Examples	30
Getting help for the command.	30
Using a parameter properties file.	30
Installing interim fixes	30
Uninstalling interim fixes	31
Viewing information about interim fixes	32
Installing fix packs	32
Uninstalling fix packs	33
Viewing information about fix packs.	33
Chapter 4. UpdateWizard command	35
Parameters.	35
Examples of use.	36
Displaying usage information	36
Bypassing the local copy of the IBM Developer Kit	36
Disabling prerequisite checking	36
Panel descriptions	36
Welcome panel	37
Product Selection panel	37
Menu panel	37
Interim Fix Repository Specifier panel	37
Installable Interim Fix Selection panel	37
Uninstallable Fix Selection panel.	38
Prerequisite Check panel	38
Pre-installation Summary and Pre-uninstallation Summary panels.	38
Installation and Uninstallation panel.	39
Post Installation Summary and Post Uninstallation Summary panels.	39
Fix Pack Repository Specifier panel.	39
Fix Pack Selection panel.	39
Fix Pack Features Selection panel	39

Pre-installation Summary and Pre-uninstallation Summary panels	40
Installation and Uninstallation panels	40
Post Installation Summary and Post Uninstallation Summary panel	40
Chapter 5. Product version and history information	41
Product information files	41
XML files in the /properties/version directory that store version information	41
XML files in the /properties/version/history directory	42
Reports	43
Product version reports	43
Product history reports	44
Logs and component backups	44
File naming convention	45
Storage locations	46
Operational description	46
Data dictionary	47
Chapter 6. Notices.	63
Third party license terms and conditions, notices and information	63
Accessing the product Web site	63
Accessing product documentation	63
Trademarks and service marks	63

About this readme_updateinstaller document

This readme_updateinstaller document describes procedures for using the update installer application to install an interim fix or fix pack on the base WebSphere Application Server product, the Network Deployment product, or the Enterprise product. This readme also describes how to remove an interim fix or fix pack from these WebSphere Application Server products.

You can also use the update installer to install interim fixes and fix packs on the WebSphere Application Server - Express product and on the WebSphere Application Server client. This readme describes using the updateSilent and the updateWizard interfaces to the update installer application. This document also describes product version and history information that the WebSphere Application Server products maintain whenever you install or remove an interim fix or fix pack.

All of the information in this document is available in the online Information Center.

Chapter 1. Installing interim fixes and fix packs

Install interim fixes and fix packs to these WebSphere Application Server products using the following procedures:

- Installing interim fixes and fix packs on the base product
- Installing interim fixes and fix packs on the Network Deployment product
- Installing interim fixes and fix packs on the Enterprise product

Use the generic procedure described in the **updateSilent** or **updateWizard** command descriptions to install interim fixes and fix packs on the WebSphere Application Server-Express product or on the WebSphere Application Server client.

Download fix packs to a read/write directory as described in the installation instructions. Installing from a read only drive is not supported at this time.

Installing interim fixes and fix packs on the base product

The update installer application installs and uninstalls interim fixes and fix packs (also known as *fixpacks*, *FixPaks* and *program temporary fixes*, or PTFs) on WebSphere Application Server products. There are two interfaces to the installer application, a wizard with a graphical interface, and a command-line, silent interface.

This topic describes the proper procedure for installing an interim fix or a fix pack on a stand-alone IBM WebSphere Application Server, V5 environment, using the update installer application.

To apply an interim fix or fix pack, first set up and configure the environment by downloading the interim fix and the update installer, or downloading the fix pack (which includes the update installer), creating update repositories, and setting the JAVA_HOME environment variable. Then use the update installer to install the interim fix or fix pack, using either its wizard interface, the **updateWizard** command, or its silent, command-line interface, the **updateSilent** command.

The update installer application can also uninstall fixes and fix packs.

If the base WebSphere Application Server node is within a cell, go to Installing interim fixes and fix packs on the Network Deployment product. If the base Application Server node is extended by the Enterprise product, go to Installing interim fixes and fix packs on the Enterprise product. This topic describes applying an interim fix or fix pack to a stand-alone base node.

Use the **versionInfo** command in the `install_root/bin` directory to display the exact fix and version level of the product. Do not use the **versionInfo** command while installing or uninstalling an interim fix or fix pack.

You can also use the silent update installer application to:

- View interim fix information
- View fix pack information

Before installing or uninstalling fixes and fix packs, stop all Java processes that use the IBM SDK that WebSphere Application Server provides to support the Java 2 SDK on your operating system platform, such as the IBM Developer Kit for AIX, Java Technology Edition. Stop all application servers, and all servers, such as the IBMHttpServer process, that belong to serviceable features. Features with servers include the IBM HTTP Server and the embedded messaging feature. Stop all Java processes, if

necessary. If you do install or uninstall an interim fix or fix pack while a WebSphere Application Server-related Java process is running, IBM does not guarantee that the product can continue to run successfully, or without error.

This procedure describes a scenario for updating a stand-alone base node by installing an interim fix or fix pack. The Uninstalling fixes and fix packs topic describes how to remove an interim fix or fix pack from a stand-alone base node.

Installing a fix pack uninstalls all interim fixes. Uninstalled fixes at a later level than the fix pack are not included in the fix pack. Reinstall such fixes to bring your system back to the previous fix level.

Space requirements vary depending on what you are installing. The size of each download is available on the Support site. For a fix pack, have approximately 400 MB of free space in the /tmp directory and another 400 MB in the file system that hosts the WebSphere Application Server image (typically /usr) on a UNIX-based platform, or approximately 800 MB of free space on the disk drive where you are installing on a Windows platform.

Verify that the free space is available before beginning the installation. After unpacking the ZIP file, you can delete the ZIP file to free space if necessary. After it is installed, the Fix Pack 2 code increases the IBM WebSphere Application Server installation and run-time footprints by a small amount.

Space is required for the /update directory of the product installation root. The space required is about the same as the size of the fix pack, typically somewhere between 50 MB to 250 MB.

Space is also required for backup files in the /properties/version/backup directory of the product installation root. The space required is about the same as the size of the fix pack, somewhere between 50 MB to 250 MB, varying by product and platform.

Interim fixes require much less space to install.

Steps for this task

1. Stop each server process on the base WebSphere Application Server node with the **stopServer** command.

Stop all WebSphere Application Server-related Java processes. On a Windows platform, use the task manager to stop Java processes. On a UNIX-based platform, use the **kill** command to stop Java processes.

2. Create an `install_root/update` directory, if it does not already exist.
3. Download the update installer application ZIP file to the `install_root/update` directory if you are installing an interim fix.

Download the current version of the file even though you might have an update installer from a previous interim fix installation. The Support page links to the current installer.

4. Extract the contents of the zip file to the update directory.
5. Create the `update/fixes` repository if you are installing an interim fix. Unpacking the fix pack creates the `fixpacks` repository directory, if it does not already exist.
6. Download the interim fix or download and unpack the fix pack.

Download an interim fix from the Support page to the `install_root/update/fixes` directory. Download a fix pack ZIP file to the `install_root/update` directory. Unpack the fix pack to automatically create the `fixpacks` directory.

On Windows platforms, the `pkunzip` utility might not decompress the download image correctly. Use another utility (such as WinZip) to unzip the image.

7. **Optional:** Set up the Java environment for the update installer.

If the update installer can set the Java environment, this step is not necessary. Otherwise, this is a required step.

The location of the update, fixes repository, and fixpacks repository directories is arbitrary. Create the directories anywhere. However, the `install_root/update`, `install_root/update/fixes`, and `install_root/update/fixpacks` locations are recommended.

If you use a non-standard installation root, it is possible that the update installer cannot set the `JAVA_HOME` environment variable. If you receive a message that the update installer cannot set `JAVA_HOME`, set the environment variable yourself, or issue the appropriate command script yourself, from the `/bin` directory of the installation root:

- a. Open a command line window.
 - b. Change directory to the `bin` directory of the installation root.
 - c. Issue the appropriate command to set `JAVA_HOME`:
 - `. install_root/bin/setupCmdLine.sh` (source the command on UNIX platforms)
 - `source install_root/bin/setupCmdLine.sh` (source the command on Linux platforms)
 - `install_root\bin\setupCmdLine.bat` (Windows platforms only)
 - `. install_root/bin/setupClient.sh` (source the command for the Application Server client)
 - `source install_root/bin/setupClient.sh` (Linux platforms only)
 - `install_root\bin\setupClient.bat` (Windows platforms only)
8. Use one of the two interfaces to the update installer to apply the interim fix or fix pack to the base node:
- Refer to **UpdateWizard** command for usage information.
 - Refer to the **UpdateSilent** command description for the proper syntax for installing the interim fix or fix pack:
 - Installing fixes
 - Installing fix packs

For example, to install the `was50_fp2_win` fix pack, use this **updateSilent** command:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fixpack
      -installDir "C:\Program Files\WebSphere\AppServer"
      -skipIHS
      -fixpackDir "C:\Program Files\WebSphere\AppServer\update\fixpacks"
      -install
      -fixpackID was50_fp2_win
```

The command is shown here on more than one line, for clarity.

9. Restart the node agent for the base node with the **startNode** command if the node is part of a cell.
- ```
startnode
```
10. Restart each server on the node with the **startServer** command.
- ```
startserver server1
```

If the node is part of a cell:

```
startserver jmsserver
```

11. Verify that the node is online and functioning correctly.

There are several ways to verify the successful application of an interim fix or fix pack:

- Does the interim fix appear in the wizard panel that lists applied interim fixes, or does the or fix pack appear in the panel that lists applied fix packs? If so, the interim fix or fix pack is installed.
- Does the interim fix or fix pack appear in the wizard panel that lists installable interim fixes, or the panel that lists installable fix packs? If so, the interim fix or fix pack is uninstalled.
- Is there an `[interim fixID].efix` or `[fix packID].ptf` file in the `install_root/properties/version/version` directory, or an `[interim fixID].efixApplied`, `[interim fixID].efixDriver`, `[fix packID].ptfApplied`, or `[fix packID].ptfDriver` file in the `install_root/properties/version/history` directory?

- Do the product version and history reports show the fix or fix pack to be installed or removed?
- Does collecting interim fix information and update state show that the interim fix is installed or removed?
- Does collecting fix pack information and update state show that the fix pack is installed or removed?

Run the installation verification tool on the node as described in the InfoCenter to verify that the node is operational.

Installing interim fixes and fix packs on the Network Deployment product

The update installer application installs and uninstalls interim fixes and fix packs (also known as *fixpacks*, *FixPaks* and *program temporary fixes*, or PTFs) on WebSphere Application Server products. There are two interfaces to the installer application, a wizard with a graphical interface, and a command-line, silent interface.

This topic describes the proper procedure for installing an interim fix or a fix pack in an IBM WebSphere Application Server Network Deployment, V5 environment, using the update installer application.

To apply an interim fix or fix pack, first set up and configure the environment by downloading the interim fix and the update installer, or downloading the fix pack (which includes the update installer), creating update repositories, and setting the `JAVA_HOME` environment variable. Then use the update installer to install the interim fix or fix pack, using either its wizard interface, the **updateWizard** command, or its silent, command-line interface, the **updateSilent** command.

The update installer application can also uninstall fixes and fix packs.

One requirement governs applying an interim fix or fix pack to a cell, to verify the continued, smooth interaction of the various WebSphere Application Server nodes:

The Network Deployment product must be at the highest fix or fix pack level within the cell.

For example, you cannot use the **addNode** command to add a V5.0.1 base WebSphere Application Server node to a V5.0.0 deployment manager cell.

There is no limitation on the interim fix or fix pack level of a base Application Server V5 node within its cell, if the fix pack level of the base node is the same as, or lower than that of the deployment manager. There is also no limit to the number of different V5 interim fix or fix pack levels that can coexist or interoperate within a cell, so long as each base node fix pack level is the same as, or lower than, that of the deployment manager.

Verify that the interim fix or fix pack level of each base WebSphere Application Server node within the cell is lower than, or identical to, the level of the deployment manager. No base node within the cell is allowed to be at a higher level than the deployment manager node. Uninstall an interim fix or fix pack from every base node if you uninstall the fix or fix pack from the deployment manager node.

Use the **versionInfo** command in the `install_root/bin` directory to display the exact interim fix and version level of the product. Do not use the **versionInfo** command while installing or uninstalling an interim fix or fix pack.

You can also use the silent update installer application to:

- View interim fix information
- View fix pack information

Before installing or uninstalling interim fixes and fix packs, stop all Java processes that use the IBM SDK that WebSphere Application Server provides to support the Java 2 SDK on your operating system platform, such as the IBM Developer Kit for AIX, Java Technology Edition. Stop all application servers, the nodeagent, the deployment manager server, and all servers, such as the jmsserver, that belong to serviceable features. Features with servers include the IBM HTTP Server and the embedded messaging feature. Stop all Java processes, if necessary. If you do install or uninstall an interim fix or fix pack while a WebSphere Application Server-related Java process is running, IBM does not guarantee that the product can continue to run successfully, or without error.

This procedure describes a scenario for updating an entire cell to the same fix pack level. According to the requirements, you can apply a fix pack to the deployment manager node only, without applying it to other nodes in the cell. Apply the fix pack to zero, one, or more of the base nodes after you apply it to the deployment manager node.

After upgrading a deployment manager node from V5 to V5.0.1 or V5.0.2, restart all node agents in the cell to verify correct operation. This includes node agents on base nodes that you have not updated to V5.0.1.

Uninstalling fixes and fix packs describes how to remove an interim fix or fix pack from an entire cell, or from any part of the cell. According to the guidelines, uninstall the fix pack from each base node in a cell before you uninstall the fix pack from the deployment manager node.

Installing a fix pack uninstalls all interim fixes. Uninstalled fixes at a later level than the fix pack are not included in the fix pack. Reinstall such fixes to bring your system back to the previous fix level.

Space requirements vary depending on what you are installing. The size of each download is available on the Support site. For a fix pack, have approximately 400 MB of free space in the /tmp directory and another 400 MB in the file system that hosts the WebSphere Application Server image (typically /usr) on a UNIX-based platform, or approximately 800 MB of free space on the disk drive where you are installing on a Windows platform.

Verify that the free space is available before beginning the installation. After unpacking the ZIP file, you can delete the ZIP file to free space if necessary. After it is installed, the Fix Pack 2 code increases the IBM WebSphere Application Server installation and run-time footprints by a small amount.

Space is required for the /update directory of the product installation root. The space required is about the same as the size of the fix pack, typically somewhere between 50 MB to 250 MB.

Space is also required for backup files in the /properties/version/backup directory of the product installation root. The space required is about the same as the size of the fix pack, somewhere between 50 MB to 250 MB, varying by product and platform.

Interim fixes require much less space to install.

Steps for this task

1. Stop the nodeagent server process on each base WebSphere Application Server node in the cell with the **stopNode** command.
The **stopNode** command is in the `install_root/bin` directory of each base node.
Stop all WebSphere Application Server-related Java processes such as the jmsserver process.
2. Stop the deployment manager process with the **stopManager** command.
The dmgr Java process is the deployment manager process. The **stopManager** command is in the `install_root/bin` directory of each base node.
3. Create an `install_root/update` directory on the network deployment node, if it does not already exist.

4. Download the update installer application ZIP file to the `install_root/update` , if you are installing an interim fix.

Download the current version of the file, even though you might have an update installer from a previous interim fix installation. The Support page links to the current installer.

5. Extract the contents of the ZIP file to the update directory.
6. Create the `update/fixes` repository if you are installing an interim fix. Unpacking the fix pack creates the `fixpacks` repository directory, if it does not already exist.
7. Download the interim fix or download and unpack the fix pack.

Download the interim fix from the Support page to the `install_root/update/fixes` repository directory, if you are installing an interim fix. Download the fix pack ZIP file to the `install_root/update` directory. Unpack the fix pack to automatically create the `/fixpacks` directory.

On Windows platforms, the `pkunzip` utility might not decompress the download image correctly. Use another utility (such as `WinZip`) to unzip the image.

8. **Optional:** Set up the Java environment for the update installer.

The location of the `update`, `fixes` repository, and `fixpacks` repository directories is arbitrary. Create the directories anywhere. However, the `install_root/update`, `install_root/update/fixes`, and `install_root/update/fixpacks` locations are recommended.

If you use a non-standard installation root, it is possible that the update installer cannot set the `JAVA_HOME` environment variable. If you receive a message that the update installer cannot set `JAVA_HOME`, set the environment variable yourself, or issue the appropriate command script yourself, from the `bin` directory of the installation root:

- a. Open a command line window.
- b. Change directories to the `bin` directory of the installation root.
- c. Run the appropriate command to set `JAVA_HOME`:
 - `. install_root/bin/setupCmdLine.sh` (source the command on UNIX platforms)
 - `source install_root/bin/setupCmdLine.sh` (source the command on Linux platforms)
 - `install_root\bin\setupCmdLine.bat` (Windows platforms only)

9. Apply the interim fix or fix pack to the deployment manager node.

Use the appropriate command to apply the interim fix or fix pack on the deployment manager node:

- Refer to **UpdateWizard** command for usage information.
- Refer to the **UpdateSilent** command description for the proper syntax for installing the interim fix or fix pack:
 - Installing fixes
 - Installing fix packs

For example, to install the `was50_nd_fp2_win` fix pack, use this **updateSilent** command:

```
C:\Program Files\WebSphere\DeploymentManager\update> updateSilent -fixpack
-installDir "C:\Program Files\WebSphere\DeploymentManager"
-skipIHS
-fixpackDir "C:\Program Files\WebSphere\DeploymentManager\update\fixpacks"
-install
-fixpackID was50_nd_fp2_win
```

The command is shown on more than one line, for clarity.

10. Bring the deployment manager node back online with the **startManager** command.

The `dmgr` Java process is the deployment manager process. The **startManager** command is in the `install_root/bin` directory of the deployment manager node.
11. Verify that the deployment manager node is fully functional and has the interim fix or fix pack applied.

There are several ways to verify the successful application of an interim fix or fix pack:

- Does the interim fix appear in the wizard panel that lists applied interim fixes, or does the or fix pack appear in the panel that lists applied fix packs? If so, the interim fix or fix pack is installed.
- Does the interim fix or fix pack appear in the wizard panel that lists installable interim fixes, or the panel that lists installable fix packs? If so, the interim fix or fix pack is uninstalled.
- Is there an [interim fixID].efix or [fix packID].ptf file in the install_root/properties/version/version directory, or an [interim fixID].efixApplied, [interim fixID].efixDriver, [fix packID].ptfApplied, or [fix packID].ptfDriver file in the install_root/properties/version/history directory?
- Do the product version reports show the fix or fix pack to be installed or removed? Do not use the **versionInfo** command while installing or uninstalling an interim fix or fix pack.
- Does collecting interim fix information and update state show that the interim fix is installed or removed?
- Does collecting fix pack information and update state show that the fix pack is installed or removed?

Run the installation verification tool on the node as described in the InfoCenter to verify that the node is operational.

12. Restart the nodeagent process on each base node with the **startNode** command.

Restart the node agent on each base node, to let the node agent continue to communicate with the updated deployment manager node. Restart all node agents if that is more convenient. Do not restart node agents on base nodes that you intend to update with the interim fix or fix pack. The interim fix or fix pack installation requires you to stop and restart the node agent.

The **startNode** command is in the install_root/bin directory of each base node.

13. Perform the base product installation for each node to which you intend to apply the interim fix or fix pack.
14. Specify that file sets on each base node match those on the deployment manager node.
Verify consistent configuration data across a cell. Synchronize files on individual nodes or throughout your system. To synchronize files throughout the system, use the Deployment Manager administrative console page, **System administration > Nodes > check_each_node_name > Full Resynchronization**. Use the administrative console page, **System Administration > Node Agents > nodeagent > File Synchronization Service** to specify automatic synchronization every minute until all base node servers are brought online.
15. Verify that all nodes are online and that the cell is functioning correctly.
16. Restore your original file synchronization settings, if you changed them.

At this point the cell is fully functional. All operations are available and function normally.

Installing interim fixes and fix packs on the Enterprise product

The update installer application installs and uninstalls interim fixes and fix packs (also known as *fixpacks*, *FixPaks* and *program temporary fixes*, or PTFs) on WebSphere Application Server products. There are two interfaces to the installer application, a wizard with a graphical interface, and a command-line, silent interface.

This topic describes the proper procedure for installing an interim fix or a fix pack on any or all Application Server products in an entire cell, or installing on a stand-alone product in an IBM WebSphere Application Server Enterprise, V5 environment, using the update installer application.

To apply an interim fix or fix pack, first set up and configure the environment by downloading the interim fix and the update installer, or downloading the fix pack (which includes the update installer), creating update repositories, and setting the JAVA_HOME environment variable. Then use the update installer to install the interim fix or fix pack, using either its wizard interface, the **updateWizard** command, or its silent, command-line interface, the **updateSilent** command.

The update installer application can also uninstall fixes and fix packs.

Three requirements govern applying an interim fix or fix pack to a cell, to verify the continued, smooth interaction of the various WebSphere Application Server products:

- **Requirement 1:** Within a cell, the Network Deployment product must be at the highest fix pack level.
For example, you cannot use the **addNode** command to add a V5.0.2 base WebSphere Application Server node to a V5.0.1 deployment manager cell.
- **Requirement 2:** The Enterprise product must be at the same fix level as the product it extends:
 - If the Enterprise product extends a base WebSphere Application Server node, the interim fix or fix pack level of the Enterprise product must be the same as that of the base WebSphere Application Server product.
 - If the Enterprise product extends a deployment manager node, the interim fix or fix pack level of the Enterprise product must be the same as that of the Network Deployment product.
- **Requirement 3:** Temporarily, while installing or removing an Enterprise product fix or fix pack, the base or Network Deployment product must be at the higher fix or fix pack level.
For example, install the interim fix or fix pack on the base product, or on the Network Deployment product before installing it on the Enterprise product. Or remove the fix or fix pack from the Enterprise product before removing it from the base product or the Network Deployment product.

Verify that the fix or fix pack level of each base WebSphere Application Server node within the cell is the same as, or lower than that of the deployment manager. No base node within the cell is allowed to be at a higher level than the deployment manager node. Uninstall an interim fix or fix pack from every base node, if you uninstall the fix or fix pack from the deployment manager node.

Use the **versionInfo** command in the `install_root/bin` directory to display the exact fix and version level of the product. Do not use the **versionInfo** command while installing or uninstalling an interim fix or fix pack.

You can also use the silent update installer application to:

- View interim fix information
- View fix pack information

Before installing or uninstalling interim fixes and fix packs on a machine, stop all Java processes on the machine that use the IBM SDK that WebSphere Application Server provides to support the Java 2 SDK on your operating system platform, such as the IBM Developer Kit for AIX, Java Technology Edition. Stop all application server processes, the nodeagent process, the deployment manager process, and all server processes, such as the jmsserver process, that belong to serviceable features. Features with server processes include the IBM HTTP Server and the embedded messaging feature. Stop all Java processes, if necessary. If you do install or uninstall an interim fix or fix pack while a WebSphere Application Server-related Java process runs, IBM does not guarantee that the product can continue to run successfully, or without error.

This procedure describes a scenario for applying an interim fix or fix pack to an Enterprise-extended base node, an entire Enterprise-extended cell, or to the Enterprise-extended deployment manager and any other Enterprise-extended base node in the cell. Apply the fix or fix pack to the Enterprise product when you install the fix or fix pack on the base or Network Deployment product. According to the guidelines:

1. Apply an interim fix or fix pack to the deployment manager.
2. Apply an interim fix or fix pack to the Enterprise product that extends the deployment manager.
3. Apply an interim fix or fix pack to zero, one, or more of the base nodes:
 - a. Apply an interim fix or fix pack to the base product on the node.
 - b. Apply an interim fix or fix pack to the Enterprise product that extends the base node.

If you are applying the interim fix or fix pack to the Enterprise product on a stand-alone application server, follow steps 8 and 9, which describe how to update a base node.

The Uninstalling fixes and fix packs topic describes how to remove an interim fix or fix pack from an Enterprise-extended base node, from an entire Enterprise-extended cell, or from any part of the cell. According to the guidelines, uninstall the interim fix or fix pack from each base node in a cell, and from the Enterprise product on each base node before uninstalling the interim fix or fix pack from the deployment manager node, and from the Enterprise product on the deployment manager node.

Installing a fix pack uninstalls all interim fixes. Uninstalled fixes at a later level than the fix pack are not included in the fix pack. Reinstall such fixes to bring your system back to the previous fix level.

Space requirements vary depending on what you are installing. The size of each download is available on the Support site. For a fix pack, have approximately 400 MB of free space in the /tmp directory and another 400 MB in the file system that hosts the WebSphere Application Server image (typically /usr) on a UNIX-based platform, or approximately 800 MB of free space on the disk drive where you are installing on a Windows platform.

Verify that the free space is available before beginning the installation. After unpacking the ZIP file, you can delete the ZIP file to free space if necessary. After it is installed, the Fix Pack 2 code increases the IBM WebSphere Application Server installation and run-time footprints by a small amount.

Space is required for the /update directory of the product installation root. The space required is about the same as the size of the fix pack, typically somewhere between 50 MB to 250 MB.

Space is also required for backup files in the /properties/version/backup directory of the product installation root. The space required is about the same as the size of the fix pack, somewhere between 50 MB to 250 MB, varying by product and platform.

Fixes require much less space to install.

Steps for this task

1. Perform the procedure in Installing interim fixes and fix packs on the Network Deployment product to update the Network Deployment node.
If you do not have a deployment manager node, skip this step.
If you do have a deployment manager node, skip the procedure to update each base node in the cell. You will do that later in this procedure

2. Stop the deployment manager process with the **stopManager** command.
The dmgr Java process is the deployment manager process. The stopManager command is in the *install_root/bin* directory of each base node.

3. Apply the same level fix or fix pack to the Enterprise product that extends the network deployment node.

After installing the fix or fix pack on the deployment manager node, apply the same level fix or fix pack to the Enterprise product that extends the network deployment node.

For example, to install the was50_pme_nd_fp2_win fix pack to the Enterprise product, use this **updateSilent** command:

```
C:\Program Files\WebSphere\DeploymentManager\update> updateSilent -fixpack
-installDir "C:\Program Files\WebSphere\DeploymentManager"
-skipIHS
-fixpackDir "C:\Program Files\WebSphere\DeploymentManager\update\fixpacks"
-install
-fixpackID was50_pme_nd_fp2_win
```

The command is shown here on more than one line, for clarity.

4. Bring the deployment manager node back online with the **startManager** command.
The dmgr Java process is the deployment manager process. The startManager command is in the *install_root/bin* directory of each base node.
5. Verify that the deployment manager node is fully functional and has the fix or fix pack applied.
There are several ways to verify the successful application of an interim fix or fix pack:
 - Does the interim fix appear in the wizard panel that lists applied interim fixes, or does the or fix pack appear in the panel that lists applied fix packs? If so, the interim fix or fix pack is installed.
 - Does the interim fix or fix pack appear in the wizard panel that lists installable interim fixes, or the panel that lists installable fix packs? If so, the interim fix or fix pack is uninstalled.
 - Is there an [interim fixID].efix or [fix packID].ptf file in the *install_root/properties/version/version* directory, or an [interim fixID].efixApplied, [interim fixID].efixDriver, [fix packID].ptfApplied, or [fix packID].ptfDriver file in the *install_root/properties/version/history* directory?
 - Do the product version reports show the fix or fix pack to be installed or removed? Do not use the **versionInfo** command while installing or uninstalling an interim fix or fix pack.
 - Does collecting interim fix information and update state show that the interim fix is installed or removed?
 - Does collecting fix pack information and update state show that the fix pack is installed or removed?
6. Stop the node agent of each base node with the **stopNode** command.
The **stopNode** command is in the *install_root/bin* directory of each base node.
7. Restart the node agent of each base node with the **startNode** command.
Restart the node agent on each base node, to let the node agent continue to communicate with the updated deployment manager node. Restart all node agents but those on base nodes that you intend to update with the interim fix or fix pack for the Enterprise product. The interim fix or fix pack installation requires you to restart the node agent later.
The **startNode** command is in the *install_root/bin* directory of each base node.
8. Perform the procedure in Installing interim fixes and fix packs on the base product to update each base node.
9. Perform the following steps for each base node to which you intend to apply the Enterprise interim fix or fix pack:
 - a. Stop each base node with the **stopNode** command.
 - b. Stop each server process on the base WebSphere Application Server node with the **stopServer** command.
Stop all WebSphere Application Server-related Java processes. On a Windows platform, use the task manager to stop Java processes. On a UNIX-based platform, use the **kill** command to stop Java processes.
 - c. Download the Enterprise interim fix or download and unpack the fix pack.
Download an interim fix from the Support page to the *install_root/update/fixes* directory. Download a fix pack ZIP file to the *install_root/update* directory. Unpack the fix pack into the *fixpacks* directory.
On Windows platforms, the pkunzip utility might not decompress the download image correctly. Use another utility (such as WinZip) to unzip the image.
 - d. Apply the interim fix or fix pack to the Enterprise product that extends the base node.
After installing the fix or fix pack to the base product, apply the same level fix or fix pack to the Enterprise product that extends the base node:
 - Refer to **UpdateWizard** command for usage information.
 - Refer to the **UpdateSilent** command description for the proper syntax for installing the interim fix or fix pack:

- Installing interim fixes
- Installing fix packs

For example, to install the was50_pme_fp2_win fix pack to the Enterprise product, use this **updateSilent** command:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fixpack
-installDir "C:\Program Files\WebSphere\AppServer"
-skipIHS
-fixpackDir "C:\Program Files\WebSphere\AppServer\update\fixpacks"
-install
-fixpackID was50_pme_fp2_win
```

The command is shown here on more than one line, for clarity.

- Restart each server on the node with the **startServer** command.
- Restart the node agent for the base node with the **startNode** command.
- Verify that the base node is fully functional and that it has the interim fix or fix pack applied.

There are several ways to verify the successful application of an interim fix or fix pack:

- Does the interim fix appear in the wizard panel that lists applied interim fixes, or does the or fix pack appear in the panel that lists applied fix packs? If so, the interim fix or fix pack is installed.
- Does the interim fix or fix pack appear in the wizard panel that lists installable interim fixes, or the panel that lists installable fix packs? If so, the interim fix or fix pack is uninstalled.
- Is there an [interim fixID].efix or [fix packID].ptf file in the install_root/properties/version/version directory, or an [interim fixID].efixApplied, [interim fixID].efixDriver, [fix packID].ptfApplied, or [fix packID].ptfDriver file in the install_root/properties/version/history directory?
- Do the product version reports show the fix or fix pack to be installed or removed? Do not use the **versionInfo** command while installing or uninstalling an interim fix or fix pack.
- Does collecting interim fix information and update state show that the interim fix is installed or removed?
- Does collecting fix pack information and update state show that the fix pack is installed or removed?

Run the installation verification tool on the node as described in the InfoCenter to verify that the node is operational.

- Specify that file sets on each base node match those on the deployment manager node.
Verify consistent configuration data across a cell. Synchronize files on individual nodes or throughout your system. To synchronize files throughout the system, use the deployment manager administrative console page, **System administration > Nodes > check_each_node_name > Full Resynchronization**. Use the administrative console page, **System Administration > Node Agents > nodeagent > File Synchronization Service** to specify automatic synchronization every minute until all base node servers are brought online.
- Verify that all nodes are online and that the cell is functioning correctly.
- Restore your original file synchronization settings, if you changed them.
The cell is now fully functional. All operations are available and functioning normally.

Chapter 2. Uninstalling fixes and fix packs

Uninstall fixes and fix packs from these WebSphere Application Server products using the following procedures:

- Removing fixes and fix packs from the base product
- Removing fixes and fix packs from the Network Deployment product
- Removing fixes and fix packs from the Enterprise product

Use the generic procedure described in the `updateSilent` or `updateWizard` commands to remove fixes and fix packs from the WebSphere Application Server-Express product or from the WebSphere Application Server client.

Removing fixes and fix packs from the base product

This topic describes the proper procedure for uninstalling an interim fix or a fix pack in an IBM WebSphere Application Server, V5 environment using the update installer application.

Removing an interim fix or fix pack requires setting the `JAVA_HOME` environment variable for the update installer. The update installer performs the task by running the `setupCmdLine` or `setupClient` command script. If you use a non-standard installation root directory for your WebSphere Application Server product, it is possible that the update installer cannot set the `JAVA_HOME` environment variable.

If the update installer throws an error because it cannot set the Java environment, set the `JAVA_HOME` variable yourself. Then you can use the update installer to uninstall the interim fix or fix pack, using either its wizard interface, the **updateWizard** command, or its silent, command-line interface, the **updateSilent** command.

The update installer application can also install fixes and fix packs.

If the base WebSphere Application Server node is within a cell, go to Removing fixes and fix packs from the Network Deployment product. If the base Application Server node is extended by the Enterprise product, go to Removing fixes and fix packs from the Enterprise product. This topic describes removing an interim fix or fix pack from a stand-alone base node.

Use the **versionInfo** command in the `install_root/bin` directory to display the exact interim fix and version level of the product. Do not use the **versionInfo** command while installing or uninstalling an interim fix or fix pack.

You can also use the silent update installer application to:

- View interim fix information
- View fix pack information

Before installing or uninstalling interim fixes and fix packs, stop all Java processes that use the IBM SDK that WebSphere Application Server provides to support the Java 2 SDK on your operating system platform, such as the IBM Developer Kit for AIX, Java Technology Edition. Stop all application servers, and all servers, such as the `IBMHttpServer` process, that belong to serviceable features. Features with servers include the IBM HTTP Server and the embedded messaging feature. Stop all Java processes, if necessary. If you do install or uninstall an interim fix or fix pack while a WebSphere Application Server-related Java process is running, IBM does not guarantee that the product can continue to run successfully, or without error.

This procedure describes a scenario for updating a stand-alone base node by removing an interim fix or fix pack. The "Installing fixes and fix packs" topic describes how to apply an interim fix or fix pack to a stand-alone base node.

Steps for this task

1. Stop each server on the base node with the **stopServer** command.
2. **(Optional)** Set up and configure your WebSphere Application Server environment.

Set up the Java environment for the update installer.

The location of the update, fixes repository, and fixpacks repository directories is arbitrary. Create the directories anywhere. However, the `install_root/update`, `install_root/update/fixes`, and `install_root/update/fixpacks` locations are recommended.

If you use a non-standard installation root, it is possible that the **updateWizard** (or **updateSilent**) command cannot set the `JAVA_HOME` environment variable. If you receive a message that the update installer cannot set `JAVA_HOME`, set the environment variable yourself, or issue the appropriate command script yourself, from the `bin` directory of the installation root:

- a. Open a command line window.
 - b. Change directories to the `bin` directory of the installation root.
 - c. Run the appropriate command:
 - `. install_root/bin/setupCmdLine.sh` (source the command on UNIX platforms)
 - `source install_root/bin/setupCmdLine.sh` (source the command on Linux platforms)
 - `install_root\bin\setupCmdLine.bat` (Windows platforms only)
 - `. install_root/bin/setupClient.sh` (source the command for the Application Server client)
 - `source install_root/bin/setupClient.sh` (Linux platforms only)
 - `install_root\bin\setupClient.bat` (Windows platforms only)
3. Use the appropriate command to remove the interim fix or fix pack from the base node:
 - Refer to **updateWizard** command topic for usage information.
 - Refer to the **updateSilent** command description for the proper syntax for uninstalling the interim fix or fix pack:
 - Uninstalling interim fixes
 - Uninstalling fix packs

For example, to uninstall the `was50_fp2_win` fix pack, use this **updateSilent** command:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fixpack
-installDir "C:\Program Files\WebSphere\AppServer"
-skipIHS
-fixpackDir "C:\Program Files\WebSphere\AppServer\update\fixpacks"
-uninstall
-fixpackID was50_fp2_win
```

The command is shown on more than one line, for clarity.

4. Restart each server on the node with the **startServer** command.
5. Verify that the node is online and functioning correctly.

There are several ways to verify the successful application of an interim fix or fix pack:

- Does the interim fix appear in the wizard panel that lists applied interim fixes, or does the or fix pack appear in the panel that lists applied fix packs? If so, the interim fix or fix pack is installed.
- Does the interim fix or fix pack appear in the wizard panel that lists installable interim fixes, or the panel that lists installable fix packs? If so, the interim fix or fix pack is uninstalled.
- Is there an `[interim fixID].efix` or `[fix packID].ptf` file in the `install_root/properties/version/version` directory, or an `[interim fixID].efixApplied`, `[interim`

`fixID].efixDriver`, `[fix packID].ptfApplied`, or `[fix packID].ptfDriver` file in the `install_root/properties/version/history` directory?

- Do the product version and history reports show the fix or fix pack to be installed or removed?
- Does collecting interim fix information and update state show that the interim fix is installed or removed?
- Does collecting fix pack information and update state show that the fix pack is installed or removed?

Run the installation verification tool on the node as described in the InfoCenter to verify that the node is operational.

Removing fixes and fix packs from the Network Deployment product

This topic describes the proper procedure for uninstalling an interim fix or a fix pack in an IBM WebSphere Application Server Network Deployment, V5 environment using the update installer application.

Removing an interim fix or fix pack requires setting the `JAVA_HOME` environment variable for the update installer. The update installer performs the task by running the `setupCmdLine` or `setupClient` command script. If you use a non-standard installation root directory for your WebSphere Application Server product, it is possible that the update installer cannot set the `JAVA_HOME` environment variable.

If the update installer throws an error because it cannot set up the Java environment, set the `JAVA_HOME` variable yourself. Then use the update installer to uninstall the interim fix or fix pack, using either its wizard interface, the **updateWizard** command, or its silent, command-line interface, the **updateSilent** command.

The update installer application can also install fixes and fix packs.

One requirement governs removing an interim fix or fix pack from a cell, to verify the continued, smooth interaction of the various WebSphere Application Server nodes: **The Network Deployment product must be at the highest interim fix or fix pack level within the cell.**

For example, you cannot use the **addNode** command of a base WebSphere Application Server node at V5.0.2, to add it to a cell that is owned by a V5.0.1 deployment manager.

There is no limitation on the fix or fix pack level of a base WebSphere Application Server V5 node within its cell, if the fix pack level of the base node is the same as, or lower than, that of the deployment manager. There is also no limit to the number of different V5 fix or fix pack levels that can coexist or interoperate within a cell, so long as each base node fix pack level is the same as, or lower than, that of the deployment manager.

Verify that the interim fix or fix pack level of each base WebSphere Application Server node within the cell, is at a lower or identical level as the deployment manager. No base node within the cell is allowed to be at a higher level than the deployment manager node. Uninstall an interim fix or fix pack from every base node, if you uninstall the interim fix or fix pack from the deployment manager node.

Use the **versionInfo** command in the `install_root/bin` directory to display the exact fix and version level of the product. Do not use the **versionInfo** command while installing or uninstalling an interim fix or fix pack.

You can also use the silent update installer application to:

- View interim fix information
- View fix pack information

Before installing or uninstalling fixes and fix packs, stop all Java processes that use the IBM SDK that WebSphere Application Server provides to support the Java 2 SDK on your operating system platform, such as the IBM Developer Kit for AIX, Java Technology Edition. Stop all application servers, the

nodeagent, the deployment manager server, and all servers, such as the jmsserver, that belong to serviceable features. Features with servers include the IBM HTTP Server and the embedded messaging feature. Stop all Java processes, if necessary. If you do install or uninstall an interim fix or fix pack while a WebSphere Application Server-related Java process is running, IBM does not guarantee that the product can continue to run successfully, or without error.

This procedure describes a scenario for removing an interim fix or fix pack from an entire Network Deployment cell, or from any part of the cell. According to the guidelines, uninstall the interim fix or fix pack from each base node in a cell before you uninstall the interim fix or fix pack from the deployment manager node.

Installing interim fixes and fix packs describes how to apply an interim fix or fix pack to an entire cell, or to selected parts of the cell.

If you plan to uninstall the Network Deployment product, uninstall all interim fixes and fix packs before uninstalling the product.

Always uninstall the highest level interim fix or fix pack before uninstalling other interim fixes or fix packs.

Steps for this task

1. Remove the highest level interim fix or fix pack from all base nodes.
Determine the base nodes from which you intend to remove the fix or fix pack. For each node perform this procedure, Removing interim fixes and fix packs from the base product.
2. Stop the deployment manager with the **stopManager** command.
3. Remove the interim fix or fix pack from the Network Deployment node.
If you remove the interim fix or fix pack from every base node in the cell, you can also remove the interim fix or fix pack from the Network Deployment node. The fix level of the Network Deployment node must be equal to, or higher than the fix level of any base node in the cell.
Use the following procedure to perform this task:
 - a. **(Optional)** Set up and configure your WebSphere Application Server environment.
Set up the Java environment for the update installer.
The location of the update, fixes repository, and fixpacks repository directories is arbitrary. Create the directories anywhere. However, the *install_root/update*, *install_root/update/fixes*, and *install_root/update/fixpacks* locations are recommended.
If you use a non-standard installation root, it is possible that the **updateWizard** (or **updateSilent**) command cannot set the JAVA_HOME environment variable. If you receive a message that the update installer cannot set JAVA_HOME, set the environment variable yourself, or issue the appropriate command script yourself, from the **bin** directory of the installation root:
 - 1) Open a command line window.
 - 2) Change directory to the bin directory of the installation root.
 - 3) Run the appropriate command:
 - `. install_root/bin/setupCmdLine.sh` (source the command on UNIX platforms)
 - `source install_root/bin/setupCmdLine.sh` (source the command on Linux platforms)
 - `install_root\bin\setupCmdLine.bat` (Windows platforms only)
 - b. Remove the fix or fix pack from the deployment manager node using either interface:
 - Refer to **updateWizard** command topic for usage information.
 - Refer to the **updateSilent** command description for the proper syntax for uninstalling the interim fix or fix pack:
 - Uninstalling interim fixes
 - Uninstalling fix packs

For example, to uninstall the was50_nd_fp2_win fix pack, use this **updateSilent** command:

```
C:\Program Files\WebSphere\DeploymentManager\update> updateSilent -fixpack
-installDir "C:\Program Files\WebSphere\DeploymentManager"
-skipIHS
-fixpackDir "C:\Program Files\WebSphere\DeploymentManager\update\fixpacks"
-uninstall
-fixpackID was50_nd_fp2_win
```

4. Start the deployment manager node with the **startManager** command.
5. Verify that the deployment manager node is fully functional and has the fix or fix pack removed.

There are several ways to verify the successful removal of an interim fix or fix pack:

- Does the interim fix appear in the wizard panel that lists applied interim fixes, or does the or fix pack appear in the panel that lists applied fix packs? If so, the interim fix or fix pack is installed.
- Does the interim fix or fix pack appear in the wizard panel that lists installable interim fixes, or the panel that lists installable fix packs? If so, the interim fix or fix pack is uninstalled.
- Is there an [interim fixID].efix or [fix packID].ptf file in the install_root/properties/version/version directory, or an [interim fixID].efixApplied, [interim fixID].efixDriver, [fix packID].ptfApplied, or [fix packID].ptfDriver file in the install_root/properties/version/history directory?
- Do the product version and history reports show the fix or fix pack to be installed or removed?
- Does collecting interim fix information and update state show that the interim fix is installed or removed?
- Does collecting fix pack information and update state show that the fix pack is installed or removed?

Run the installation verification tool on the node as described in the InfoCenter to verify that the node is operational.

6. Specify that file sets on one node match those on the deployment manager node.
Verify consistent configuration data across a cell. Synchronize files on individual nodes or throughout your system. To synchronize files throughout the system, use the Deployment Manager administrative console page, **System administration > Nodes > check_each_node_name > Full Resynchronization**. Use the administrative console page, **System Administration > Node Agents > nodeagent > File Synchronization Service** to specify automatic synchronization every minute until all base node servers are brought online.
7. Verify that all nodes are online and that the cell is functioning correctly.
8. Restore your original file synchronization settings.
At this point the cell is fully functional. All operations are available and function normally.

Removing fixes and fix packs from the Enterprise product

This topic describes the proper procedure for uninstalling an interim fix or a fix pack on an entire cell or on a single machine in an IBM WebSphere Application Server Enterprise, V5 environment using the update installer application.

Removing an interim fix or fix pack requires setting the JAVA_HOME environment variable for the update installer. The update installer performs the task by running the **setupCmdLine** or **setupClient** command script. If you use a non-standard installation root directory for your WebSphere Application Server product, it is possible that the update installer cannot set the JAVA_HOME environment variable.

If the update installer throws an error because it cannot set up the Java environment, set the JAVA_HOME variable yourself. Then you can use the update installer to uninstall the interim fix or fix pack, using either its wizard interface, the **updateWizard** command, or its silent, command-line interface, the **updateSilent** command.

The update installer application can also install fixes and fix packs.

Three requirements govern removing an interim fix or fix pack to a cell, to verify the continued, smooth interaction of the various WebSphere Application Server products:

- **Requirement 1:** Within a cell, the Network Deployment product must be at the highest interim fix or fix pack level.

For example, you cannot use the **addNode** command to add a V5.0.2 base WebSphere Application Server node to a V5.0.1 deployment manager cell.

- **Requirement 2:** The Enterprise product must be at the same interim fix or fix pack level as the product it extends:

- If the Enterprise product extends a base WebSphere Application Server node, the interim fix or fix pack level of the Enterprise product must be the same as that of the base WebSphere Application Server product.
- If the Enterprise product extends a deployment manager node, the interim fix or fix pack level of the Enterprise product must be the same as that of the Network Deployment product.

- **Requirement 3:** Temporarily, while installing or removing an Enterprise product interim fix or fix pack, the base or Network Deployment product must be at the higher interim fix or fix pack level.

For example, install the interim fix or fix pack on the base product, or on the Network Deployment product before installing it on the Enterprise product. Or remove the interim fix or fix pack from the Enterprise product before removing it from the base product or the Network Deployment product.

Before installing or uninstalling fixes and fix packs on a machine, stop all Java processes on the machine that use the IBM SDK that WebSphere Application Server provides to support the Java 2 SDK on your operating system platform, such as the IBM Developer Kit for AIX, Java Technology Edition. Stop all application server processes, the nodeagent process, the deployment manager process, and all server processes, such as the jmsserver process, that belong to serviceable features. Features with server processes include the IBM HTTP Server and the embedded messaging feature. Stop all Java processes, if necessary. If you do install or uninstall an interim fix or fix pack while a WebSphere Application Server-related Java process runs, IBM does not guarantee that the product can continue to run successfully, or without error.

This procedure describes a scenario for removing an interim fix or fix pack from a base node, from an entire cell, or from any part of the cell in a WebSphere Application Server Enterprise environment. According to the guidelines, uninstall the fix or fix pack from each base node in a cell before you uninstall the fix or fix pack from the deployment manager node. Remove the fix or fix pack from the Enterprise product when you remove it from the base or Network Deployment product.

The Installing fixes and fix packs topic describes how to apply an interim fix or fix pack to an entire cell, or to selected parts of the cell.

Steps for this task

1. Remove the Enterprise interim fix or fix pack from a base node as described in Removing fixes and fix packs from the base product, but remove the Enterprise interim fix or fix pack.

For example, to remove the was50_pme_fp2_win fix pack from the Enterprise product, use this **updateSilent** command:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fixpack
-installDir "C:\Program Files\WebSphere\AppServer"
-skipIHS
-fixpackDir "C:\Program Files\WebSphere\AppServer\update\fixpacks"
-uninstall
-fixpackID was50_pme_fp2_win
```

The command is shown here on more than one line, for clarity.

2. Remove the base product interim fix or fix pack from the base node, if necessary, as described in Removing fixes and fix packs from the base product.

If you do not have a deployment manager node, you are finished with this procedure.

Note: If you remove the interim fix or fix pack from every base node in the cell, you can also remove the interim fix or fix pack from the Network Deployment node. The fix level of the Network Deployment node must be equal to or higher than the fix level of any base node in the cell.

3. Remove the Enterprise fix or fix pack from the Network Deployment node, as described in Removing fixes and fix packs from the Network Deployment product, but remove the Enterprise interim fix or fix pack.

For example, to remove the was50_pme_nd_fp2_win fix pack from the Enterprise product, use this **updateSilent** command:

```
C:\Program Files\WebSphere\DeploymentManager\update> updateSilent -fixpack
-installDir "C:\Program Files\WebSphere\DeploymentManager"
-skipIHS
-fixpackDir "C:\Program Files\WebSphere\DeploymentManager\update\fixpacks"
-uninstall
-fixpackID was50_pme_nd_fp2_win
```

The command is shown here on more than one line, for clarity.

4. Remove the fix or fix pack from the Network Deployment node, if necessary, as described in Removing fixes and fix packs from the Network Deployment product.

For example, to uninstall the was50_nd_fp2_win fix pack, use this **updateSilent** command:

```
C:\Program Files\WebSphere\DeploymentManager\update> updateSilent -fixpack
-installDir "C:\Program Files\WebSphere\DeploymentManager"
-skipIHS
-fixpackDir "C:\Program Files\WebSphere\DeploymentManager\update\fixpacks"
-uninstall
-fixpackID was50_nd_fp2_win
```

The command is shown here on more than one line, for clarity.

Verify that the interim fix or fix pack level of each base WebSphere Application Server node within the cell is the same as, or lower than that of the deployment manager. No base node within the cell is allowed to be at a higher level than the deployment manager node. Uninstall an interim fix or fix pack from every base node, if you uninstall the fix or fix pack from the deployment manager node.

According to the guidelines:

1. Remove an interim fix or fix pack from all base nodes:
 - a. Remove the interim fix or fix pack from the Enterprise product that extends the base node.
 - b. Remove the interim fix or fix pack from the base product on the node.
2. Remove the interim fix or fix pack from the Enterprise product that extends the deployment manager.
3. Remove the interim fix or fix pack from the deployment manager.

Use the **versionInfo** command in the `install_root/bin` directory to display the exact fix and version level of the product. Do not use the **versionInfo** command while installing or uninstalling an interim fix or a fix pack.

You can also use the silent update installer application to:

- View interim fix information
- View fix pack information

Chapter 3. UpdateSilent command

The **updateSilent** command is the silent, command-line interface to the IBM WebSphere Application Server update installer application. You can also use a wizard interface to the update installer application, the **updateWizard** command. The update installer installs and uninstalls fixes and fix packs for WebSphere Application Server products. This topic describes the silent interface to the update installer command, and its command-line parameters.

Stop all Java processes on the machine that use the IBM SDK that WebSphere Application Server provides Before installing or uninstalling fixes and fix packs on a machine, stop all Java processes on the machine that use the IBM SDK that WebSphere Application Server provides to support the Java 2 SDK on your operating system platform, such as the IBM Developer Kit for AIX, Java Technology Edition. Stop all application server processes, the nodeagent process, the deployment manager process, and all server processes, such as the jmsserver process, that belong to serviceable features. Features with server processes include the IBM HTTP Server and the embedded messaging feature. Stop all Java processes, if necessary. If you do install or uninstall an interim fix or fix pack while a WebSphere Application Server-related Java process runs, IBM does not guarantee that the product can continue to run successfully, or without error.

Remove the WebSphere MQ tray icon if present On a Windows platform, remove the WebSphere MQ tray icon if it is present. The WebSphere MQ tray icon in the lower right corner indicates that a WebSphere MQ process (amqmtbrn.exe) is running. Right click the tray icon and click Hide to remove it.

Do not launch multiple copies of the update installer at one time The update installer cannot be launched concurrently with itself. Performing more than one update at the same time can lead to a failed or faulty installation.

Installation roots

The symbol *install_root* means the root directory for WebSphere Application Server. By default, this varies per product and operating system:

- Base WebSphere Application Server product:
 - AIX platforms: /usr/WebSphere/AppServer
 - Other UNIX and Linux platforms: /opt/WebSphere/AppServer
 - Windows platforms: *drive*\Program Files\WebSphere\AppServer
- Network Deployment product:
 - AIX platforms: /usr/WebSphere/DeploymentManager
 - Other UNIX and Linux platforms: /opt/WebSphere/DeploymentManager
 - Windows platforms: *drive*\Program Files\WebSphere\ DeploymentManager
- Enterprise product that extends the base product:
 - AIX platforms: /usr/WebSphere/AppServer
 - Other UNIX and Linux platforms: /opt/WebSphere/AppServer
 - Windows platforms: *drive*\Program Files\WebSphere\AppServer
- Enterprise product that extends the Network Deployment product
 - AIX platforms: /usr/WebSphere/DeploymentManager
 - Other UNIX and Linux platforms: /opt/WebSphere/DeploymentManager
 - Windows platforms: *drive*\Program Files\WebSphere\DeploymentManager

Space requirements

Space requirements vary depending on what you are installing. The size of each download is available on the Support site. After unpacking the ZIP file you download, delete the ZIP file to free space. For a fix pack, have approximately 400 MB of free space in the /tmp directory and another

400 MB in the file system that hosts the WebSphere Application Server image (typically /usr) on a UNIX-based platform, or approximately 800 MB of free space on the disk drive where you are installing on a Windows platform.

Verify that the free space is available before beginning the installation. After unpacking the ZIP file, you can delete the ZIP file to free space if necessary. After it is installed, the Fix Pack 2 code increases the IBM WebSphere Application Server installation and run-time footprints by a small amount.

Space is also required for backup files in the `install_root/properties/version/backup` directory. When installing a fix pack the space required is typically about the same as the size of the fix pack, that is, between 50 MB and 250 MB, depending on the particular fix pack.

Fixes require much less space to install.

The update installer checks for required space before it installs the fix pack. However, there is a bug in the space checker when installing Fix Pack 2.

When installing Fix Pack 2, the update installer checks for required disk space in the `/tmp` directory and in the backup directory. If there is not enough space in the `/tmp` directory or there is not enough space in the backup directory, the update installer issues a warning message.

A temporary problem exists where the update installer issues the warning for both directories when it should issue a warning message for the `/tmp` directory only. For example, if there is less than 248 MB in the `/tmp` directory and more than 248 MB in the installation root, messages similar to these appear:

```
"Insufficient disc space found in /tmp.  
This fix pack installation requires 248 megabytes."
```

```
"Insufficient disc space found in /WebSphere/AppServer/properties/version/backup.  
This fix pack installation requires 248 megabytes."
```

To fix the problem, allocate enough space in the `/tmp` directory. Neither message will appear.

Command name

`updateSilent.sh` and `updateSilent.bat`, command-line interface to the `installer.jar` file.

Related command

`updateWizard.sh` and `updateWizard.bat`, graphical interface to the `installer.jar` file.

Prerequisite environment setting

The `JAVA_HOME` environment setting.

If you use a non-standard installation root, it is possible that the **updateWizard** command cannot set the `JAVA_HOME` environment variable. If you receive a message that the wizard is unable to set `JAVA_HOME`, set the environment variable yourself, or issue the appropriate command script yourself, from the `/bin` directory of the installation root:

1. Open a command line window.
2. Change directories to the `bin` directory of the `install_root`.
3. Run the appropriate command:
 - `. install_root/bin/setupCmdLine.sh` (source the command on UNIX platforms)
 - `source install_root/bin/setupCmdLine.sh` (source the command on Linux platforms)
 - `install_root\bin\setupCmdLine.bat` (Windows platforms only)
 - `. install_root/bin/setupClient.sh` (source the command for the Application Server client)
 - `source install_root/bin/setupClient.sh` (Linux platforms only)
 - `install_root\bin\setupClient.bat` (Windows platforms only)

Download from

Download as `updateInstaller.zip` from the WebSphere Application Server Support page, or as

part of each fix pack ZIP file package. Fix packs are named according to the Application Server product, the fix pack, and the operating system platform:

Table 1. Fix pack names for Fix Pack 1. (The spaces in each file name after each underscore are for print formatting purposes.)

Product	Operating system platform	Fix Pack 2 ZIP file	Fix Pack 2 ID	Default repository in installation root directory
Base WebSphere Application Server	AIX	was50_fp2_aix.zip	was50_fp2_aix	../update/ fixpacks
	HP-UX	was50_fp2_hpux.zip	was50_fp2_hpux	
	Linux	was50_fp2_linux.zip	was50_fp2_linux	
	Linux for S/390	was50_fp2_linux390.zip	was50_fp2_linux390	
	Solaris	was50_fp2_solaris.zip	was50_fp2_solaris	
	Windows	was50_fp2_win.zip	was50_fp2_win	..\update\ fixpacks
Network Deployment	AIX	was50_nd_fp2_aix.zip	was50_nd_fp2_aix	Move the fix pack to a unique directory, such as ../update/fixpacks/ nd, to improve performance when there is a base fix pack in the default directory.
	HP-UX	was50_nd_fp2_hpux.zip	was50_nd_fp2_hpux	
	Linux	was50_nd_fp2_linux.zip	was50_nd_fp2_linux	
	Linux for S/390	was50_nd_fp2_linux390.zip	was50_nd_fp2_linux390	
	Solaris	was50_nd_fp2_solaris.zip	was50_nd_fp2_solaris	
	Windows	was50_nd_fp2_win.zip	was50_nd_fp2_win	..\update\ fixpacks

Table 1. Fix pack names for Fix Pack 1 (continued). (The spaces in each file name after each underscore are for print formatting purposes.)

Product	Operating system platform	Fix Pack 2 ZIP file	Fix Pack 2 ID	Default repository in installation root directory
Enterprise	AIX	was50_pme_fp2_aix.zip	was50_pme_fp2_aix (to extend the base product)	../update/ fixpacks Move each fix pack to a unique directory, such as ../update/fixpacks/ent and ../update/fixpacks/ent/nd, to improve performance if there is another fix pack in the default directory.
			was50_pme_nd_fp2_aix (to extend the Network Deployment product)	
	HP-UX	was50_pme_fp2_hpux.zip	was50_pme_fp2_hpux (to extend the base product)	
			was50_pme_nd_fp2_hpux (to extend the base product)	
	Linux	was50_pme_fp2_linux.zip	was50_pme_fp2_linux (base)	
			was50_pme_nd_fp2_linux (Network Deployment)	
	Linux for S/390	was50_pme_fp2_linux390.zip	was50_pme_fp2_linux390 (base)	
			was50_pme_nd_fp2_linux390 (Network Deployment)	
	Solaris	was50_pme_fp2_solaris.zip	was50_pme_fp2_solaris (base)	
			was50_pme_nd_fp2_solaris (Network Deployment)	
	Windows	was50_pme_fp2_win.zip	was50_pme_fp2_win (base)	..\update\ fixpacks
			was50_pme_nd_fp2_win (Network Deployment)	
Express	AIX	was50_express_fp2_aix.zip	was50_express_fp2_aix	../update/ fixpacks
	HP-UX	was50_express_fp2_hpux.zip	was50_express_fp2_hpux	Move the fix pack to a unique directory, such as ../update/fixpacks/express, to improve performance if there is another fix pack in the default directory.
	Linux	was50_express_fp2_linux.zip	was50_express_fp2_linux	
	Linux for S/390	was50_express_fp2_linux390.zip	was50_express_fp2_linux390	
	Solaris	was50_express_fp2_solaris.zip	was50_express_fp2_solaris	
	Windows	was50_express_fp2_win.zip	was50_express_fp2_win	..\update\ fixpacks

Table 1. Fix pack names for Fix Pack 1 (continued). (The spaces in each file name after each underscore are for print formatting purposes.)

Product	Operating system platform	Fix Pack 2 ZIP file	Fix Pack 2 ID	Default repository in installation root directory
WebSphere Application Server client	AIX	was50_client_fp2_aix.zip	was50_client_fp2_aix	../update/ fixpacks
	HP-UX	was50_client_fp2_hpux.zip	was50_client_fp2_hpux	Move the fix pack to a unique directory, such as ../update/ fixpacks/ client, to improve performance if there is another fix pack in the default directory.
	Linux	was50_client_fp2_linux.zip	was50_client_fp2_linux	
	Linux for S/390	was50_client_fp2_linux390.zip	was50_client_fp2_linux390	
	Solaris	was50_client_fp2_solaris.zip	was50_client_fp2_solaris	
	Windows	was50_client_fp2_win.zip	was50_client_fp2_win	..\update\ fixpacks

Download to

The default location for unpacking the update installer or fix pack zip file is the `install_root/update` directory. Unpacking a fix pack creates the `../update/fixpacks` directory. Create another directory, `../update/fixes`, for a repository of fixes you download. If you use these default subdirectories, accept default interim fix and fix pack file locations when using the **updateWizard** interface. Otherwise, browse to locate the interim fixes or fix packs you are installing or uninstalling.

Create an update directory if it does not exist, download the update installer application ZIP file from the Support site to the update directory, and unzip the file.

On Windows platforms, the `pkunzip` utility might not decompress the download image correctly. Use another utility (such as WinZip) to unzip the image.

Location of extfile.jar

WebSphere Application Server product `install_root/update/lib` (or `install_root\update\lib` for Windows platforms)

Location of installer.jar, readme_ptf.html, updateSilent.sh/bat, and updateWizard.sh/bat

WebSphere Application Server product `install_root/update` (or `install_root\update` for Windows platforms)

Location of fix Java archive (JAR) files

`../update/fixes` (or `..\update\fixes`)

Location of fix pack JAR files

`../update/fixpacks` (or `..\update\fixpacks`)

This directory is the location after unpacking the fix pack in the `install_root/update` directory.

Files in updateInstaller.zip

Always use the **updateSilent** (or **updateWizard**) command file from the `updateInstaller.zip` or fix pack you download, to use the most recent version. Files in the `updateInstaller.zip` package (or the fix pack ZIP package) include:

- `extfile.jar`
- `installer.jar`
- `readme_updateinstaller.txt`
- `readme_updateinstaller.html`

- readme_updateinstaller.pdf
- updateSilent.sh (or updateSilent.bat)
- updateWizard.sh (or updateWizard.bat)

In addition to the files listed, the fix pack ZIP file also has the fix pack JAR file, such as the was50_fp2_win.jar file. Each JAR file includes a fix pack.

Location of log and backup files

The update installer records processing results in log files in theinstall_root/properties/version/log directory. Backup files created during the installation of interim fixes and fix packs are in the install_root/properties/version/backup directory. The files are required to uninstall an interim fix or fix pack.

Beginning with Fix Pack 2, the update installer log files are in theinstall_root/logs/update directory.

Syntax examples

The silent update application actually provides two functions. Depending upon the parameters you choose, the command:

- Installs and uninstalls interim fixes and fix packs
- Provides information about the update state of interim fixes and fix packs you apply

The following examples describe the syntax of various uses of the update installer. In each example, optional parameters are enclosed by brackets ([]). Values that you supply appear in *italicized font*. Choices are denoted by the pipe symbol (|).

Help

```
updateSilent -help | -? | /help | -usage
```

Use a properties file to supply values

```
updateSilent myProps.properties
```

Fix processing

```
updateSilent -installDir "fully qualified product install_root"
               -fix
               -fixDir "fully qualified fix repository root,
                       usually install_root/update/fixes"
               -install | -uninstall | uninstallAll
               -fixes space-delimited list of fixes
               -fixJars space-delimited list of fix JAR files
               [-fixDetails]
               [-prereqOverride]
```

View applied fixes

```
updateSilent -fix
               -installDir "fully qualified product install_root"
```


View available fixes

```
updateSilent -fix
              -installDir "fully qualified product install_root"
              -fixDir "fully qualified fix repository root,
                      usually install_root/update/fixes"
```

Fix pack processing

```
updateSilent -installDir "fully qualified product install_root"
              -fixpack
              -fixpackDir "fully qualified FixPack repository root,
                          usually install_root/update/fixpacks"
              -install | -uninstall
              -fixPackID fix pack ID
              [-skipIHS | [-ihsOnly] -ihsInstallDir fully qualified IBM HTTP Server root]
              [-skipMQ | -mqInstallDir embedded messaging feature root]
              [-includeOptional space-delimited list of components]
              [-fixpackDetails]
```

All other valid arguments are ignored, such as the `prereqOverride` argument, which is for fix processing only.

You need not supply the `-mqInstallDir` argument for AIX, Linux, and UNIX-based platforms. The install location is fixed on those operating platforms. Use the argument on Windows platforms. The default location on Windows platforms is the `C:\Program Files\IBM\WebSphere MQ` directory.

View applied fix packs

```
updateSilent -fixpack
              -installDir "fully qualified product install_root"
```

View available fix packs

```
updateSilent -fixpack
              -installDir "fully qualified product install_root"
              -fixpackDir "fully qualified fix pack repository root,
                          usually install_root/update/fixpacks"
```

Parameters

Use the following parameters for the **updateSilent** command:

- ?** Shows command usage.
- /?** Shows command usage on Windows platforms only. Not supported for Linux and UNIX-based platforms.
- fix** Interim fix only: Identifies the update as an interim fix update.
- fixDetails** Interim fix only: Displays interim fix detail information.
- fixDir** Interim fix only: Specifies the fully qualified directory where you download interim fixes. This directory is usually the `install_root/update/fixes` directory.
- fixes** Interim fix only: Specifies a list of space-delimited interim fixes to install or uninstall.
- fixJars** Interim fix only: Specifies a list of space-delimited interim fix JAR files to install or uninstall. Each JAR file has one or more interim fixes.

-fixpack

Fix pack only: Identifies the update as a fix pack update.

-fixpackDetails

Fix pack only: Displays fix pack detail information.

-fixpackDir

Fix pack only: Specifies the fully qualified directory where you download and unpack fix packs. By default, this directory is the *install_root/update/fixpacks* directory.

-fixpackID

Fix pack only: Specifies the ID of a fix pack to install or uninstall. The value you specify does not include the *.jar* extension. The value is not the fully qualified package file name, but is the name of the individual fix pack within the JAR file.

The current Application Server strategy for fix pack JAR files is to use one JAR file per fix pack. The fix pack ID is the name of the JAR file before the *.jar* extension. For example:

- **fix pack ID:** was50_fp2_linux
- **fix pack JAR file name:** was50_fp2_linux.jar
- **fix pack ZIP file name:** was50_fp2_linux.zip

-help Shows command usage.

/help Shows command usage.

-ihsInstallDir

Fix pack only: Specifies the fully qualified path of the IBM HTTP Server product, and applies any service for the IBM HTTP Server product that might exist in the fix pack.

-ihsOnly

Fix pack only: Skips the installation of all service but that for the IBM HTTP Server product, and applies any service for the IBM HTTP Server product that might exist in the fix pack. Requires the *-ihsInstallDir* parameter.

-includeOptional

Fix pack only: Specifies a space-delimited list of features. The installer applies any service for the components, if present in the fix pack. Otherwise, the installer does not apply the service.

-install

Installs the update type.

-installDir

Specifies the fully qualified installation root of the WebSphere Application Server product.

-mqInstallDir

Fix pack only: Specifies the fully qualified installation root of the embedded messaging feature, which is based on WebSphere MQ technology.

-prereqOverride

Interim fix only: Overrides any installation and uninstallation prerequisite checking. The update installer does not log missing prerequisites.

<propertyFile>.properties

Specifies an externally supplied parameters file.

You can supply parameters in an external *.properties* file, rather than directly on the command line. There are some differences in the formats of parameters:

- Parameters are [name]=[value] pairs.
- Lists of parameter values are comma-delimited instead of space-delimited.
- There are two slashes before directory names.

Use the *.properties* file template included as part of the update installer download.

For example, a sample.properties file for installing two fixes might look like this:

```
#Sample.properties
#Sample parameters file to install fixes with details and prerequisite override
fix=true
install=true
installDir=C:\\WebSphere\\AppServer
fixDir=C:\\WebSphere\\AppServer\\update\\fixes
fixes=Fix1,Fix2
fixDetails=true
prereqOverride=true
```

A sample.properties file for installing a fix pack might look like this:

```
#Sample.properties
#Sample parameters file to install a fix pack with details
install=true
installDir=C:\\WebSphere\\AppServer
fixpackDir=C:\\WebSphere\\AppServer\\update\\fixpacks
fixpackID=was50_fp2_win
fixpackDetails=true
ihsInstallDir=C:\\IBMHttpServer
```

-skipIHS

Fix pack only: Skips any optional service for the IBM HTTP Server product that might exist in the fix pack.

If you installed the IBM HTTP Server product as a feature, use the update installer to update it with service in an interim fix or fix pack. Otherwise, download an updated IBM HTTP Server product and install it into the same directory as your existing version, to update the existing installation. You can also uninstall the current version and install the downloaded version to avoid any issues with migration.

Update your configuration if you reinstall. The process is described in the *Manually configuring supported Web servers* (tins_manualWebserver) topic in the InfoCenter for the base Application Server.

-skipMQ

Fix pack only: Skips any optional service for the embedded messaging feature (which is based on the IBM WebSphere MQ product) that might exist in the fix pack.

Always apply any outstanding corrective service to the stand-alone IBM WebSphere MQ product if you have it, before using the WebSphere Application Server update installer to update the embedded messaging feature with service in an interim fix or fix pack. Skip the installation of service to the embedded messaging feature if you install corrective service to the stand-alone IBM WebSphere MQ product.

-uninstall

Specifies to uninstall the identified fix or fix pack.

-uninstallAll

Interim fix only: Specifies to uninstall all applied interim fixes. This parameter does not uninstall fix packs.

You must uninstall all interim fixes and fix packs before uninstalling the base WebSphere Application Server product, the Network Deployment product, or the Enterprise product.

-usage

Shows command usage.

Examples

The following examples assume that:

- The installation root is the C:\Program Files\WebSphere\AppServer directory.
- The location of the IBM HTTP Server feature is the C:\Program Files\IBMHttpServer directory.
- The fix repository is the C:\Program Files\WebSphere\AppServer\update\fixes directory.
- The fix pack repository is the C:\Program Files\WebSphere\AppServer\update\fixpacks directory.

Examples in this section include:

- Getting help for the command
- Using a parameter properties file
- Installing interim fixes
- Uninstalling interim fixes
- Viewing information about interim fixes
- Installing fix packs
- Uninstalling fix packs
- Viewing information about fix packs

Most of the examples are split into more than one line, for clarity.

Getting help for the command

To get help for the **updateSilent** command:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -help
```

Using a parameter properties file

To use the myProps.properties file to supply parameter values for the **updateSilent** command:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent myProps.properties
```

Installing interim fixes

To install a collection of interim fixes:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fix
    -installDir "C:\Program Files\WebSphere\AppServer"
    -fixDir "C:\Program Files\WebSphere\AppServer\update\fixes"
    -install
    -fixes Fix1 Fix2
```

To install a collection of interim fixes, and display interim fix details:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fix
    -installDir "C:\Program Files\WebSphere\AppServer"
    -fixDir "C:\Program Files\WebSphere\AppServer\update\fixes"
    -install
    -fixes Fix1 Fix2
    -fixDetails
```

To install a collection of fixes, and override prerequisite checking:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fix
    -installDir "C:\Program Files\WebSphere\AppServer"
    -fixDir "C:\Program Files\WebSphere\AppServer\update\fixes"
    -install
    -fixes Fix1 Fix2
    -prereqOverride
```

To install interim fixes from a Java archive (JAR) file:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fix
-installDir "C:\Program Files\WebSphere\AppServer"
-fixDir "C:\Program Files\WebSphere\AppServer\update\fixes"
-install
-fixJar Fix1
```

To install interim fixes from a Java archive (JAR) file, and display interim fix details:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fix
-installDir "C:\Program Files\WebSphere\AppServer"
-fixDir "C:\Program Files\WebSphere\AppServer\update\fixes"
-install
-fixJar Fix1
-fixDetails
```

To install interim fixes from a Java archive (JAR) file, and override prerequisite checking:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fix
-installDir "C:\Program Files\WebSphere\AppServer"
-fixDir "C:\Program Files\WebSphere\AppServer\update\fixes"
-install
-fixJar Fix1
-fixDetails
```

Uninstalling interim fixes

To uninstall a collection of interim fixes:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fix
-installDir "C:\Program Files\WebSphere\AppServer"
-fixDir "C:\Program Files\WebSphere\AppServer\update\fixes"
-uninstall
-fixes Fix1 Fix2
```

To uninstall a collection of interim fixes, and display interim fix details:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fix
-installDir "C:\Program Files\WebSphere\AppServer"
-fixDir "C:\Program Files\WebSphere\AppServer\update\fixes"
-uninstall
-fixes Fix1 Fix2
-fixDetails
```

To uninstall a collection of interim fixes, and override prerequisite checking:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fix
-installDir "C:\Program Files\WebSphere\AppServer"
-fixDir "C:\Program Files\WebSphere\AppServer\update\fixes"
-uninstall
-fixes Fix1 Fix2
-prereqOverride
```

To uninstall interim fixes in a Java archive (JAR) file:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fix
-installDir "C:\Program Files\WebSphere\AppServer"
-fixDir "C:\Program Files\WebSphere\AppServer\update\fixes"
-uninstall
-fixJar Fix1
```

To uninstall interim fixes in a Java archive (JAR) file, and display interim fix details:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fix
-installDir "C:\Program Files\WebSphere\AppServer"
-fixDir "C:\Program Files\WebSphere\AppServer\update\fixes"
-uninstall
-fixJar Fix1
-fixDetails
```

To uninstall interim fixes in a Java archive (JAR) file, and override prerequisite checking:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fix
    -installDir "C:\Program Files\WebSphere\AppServer"
    -fixDir "C:\Program Files\WebSphere\AppServer\update\fixes"
    -uninstall
    -fixJar Fix1
    -fixDetails
```

Viewing information about interim fixes

To view a list of installed interim fixes:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fix
    -installDir "C:\Program Files\WebSphere\AppServer"
```

To view a list of interim fixes available in the repository:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fix
    -installDir "C:\Program Files\WebSphere\AppServer"
    -fixDir "C:\Program Files\WebSphere\AppServer\update\fixes"
```

Installing fix packs

To install a fix pack:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fixpack
    -installDir "C:\Program Files\WebSphere\AppServer"
    -ihsInstallDir "C:\Program Files\IBMHttpServer"
    -fixpackDir "C:\Program Files\WebSphere\AppServer\update\fixpacks"
    -install
    -fixpackID was50_fp2_win
```

To install a fix pack, and display fix pack details:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fixpack
    -installDir "C:\Program Files\WebSphere\AppServer"
    -ihsInstallDir "C:\IBMHttpServer"
    -fixpackDir "C:\Program Files\WebSphere\AppServer\update\fixpacks"
    -install
    -fixpackID was50_fp2_win
    -fixpackDetails
```

To perform a partial installation of a fix pack, by choosing to skip the installation of optional service to the WebSphere Application Server embedded messaging feature, which is based on WebSphere MQ:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fixpack
    -installDir "C:\Program Files\WebSphere\AppServer"
    -fixpackDir "C:\Program Files\WebSphere\AppServer\update\fixpacks"
    -ihsInstallDir "C:\Program Files\IBMHttpServer"
    -install
    -fixpackID was50_fp2_win
    -skipMQ
```

The fix pack status shows partial installation.

To perform a partial installation of a fix pack, by choosing to skip the installation of optional service to the IBM HTTP Server feature:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fixpack
    -installDir "C:\Program Files\WebSphere\AppServer"
    -fixpackDir "C:\Program Files\WebSphere\AppServer\update\fixpacks"
    -mqInstallDir "C:\WebSphere MQ"
    -install
    -fixpackID was50_fp2_win
    -skipIHS
```

The fix pack status shows partial installation.

To perform a partial installation of a fix pack, by choosing to skip the installation of optional service to both the embedded messaging feature and the IBM HTTP Server feature:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fixpack
               -installDir "C:\Program Files\WebSphere\AppServer"
               -fixpackDir "C:\Program Files\WebSphere\AppServer\update\fixpacks"
               -install
               -fixpackID was50_fp2_win
               -skipIHS
               -skipMQ
```

The fix pack status shows partial installation.

Uninstalling fix packs

To uninstall a fix pack:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fixpack
               -installDir "C:\Program Files\WebSphere\AppServer"
               -uninstall
               -fixpackID was50_fp2_win
```

To uninstall a fix pack, and display fix pack details:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fixpack
               -installDir "C:\Program Files\WebSphere\AppServer"
               -uninstall
               -fixpackID was50_fp2_win
               -fixpackDetails
```

Viewing information about fix packs

To view a list of installed fix packs:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fixpack
               -installDir "C:\Program Files\WebSphere\AppServer"
```

To view a list of fix packs available in the repository for the base WebSphere Application Server product:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fixpack
               -installDir "C:\Program Files\WebSphere\AppServer"
               -fixpackDir "C:\Program Files\WebSphere\AppServer\update\fixpacks"
```

Chapter 4. UpdateWizard command

The **updateWizard** command is the wizard interface to the IBM WebSphere Application Server update installer application. You can also use a silent, command-line interface to the update installer application, the **updateSilent** command. The update installer installs and uninstalls interim fixes and fix packs for WebSphere Application Server products.

This topic describes the wizard interface to the update installer and gives information about its panels and fields.

Before installing or uninstalling interim fixes and fix packs, stop all Java processes that use the IBM SDK that WebSphere Application Server provides to support the Java 2 SDK on your operating system platform, such as the IBM Developer Kit for AIX, Java Technology Edition. Stop all application server processes, the nodeagent process, the deployment manager process, and all server processes, such as the jmsserver process, that belong to serviceable features. Features with server processes include the IBM HTTP Server and the embedded messaging feature.

Stop all Java processes, if necessary. If you do install or uninstall an interim fix or fix pack while a WebSphere Application Server-related Java process is running, IBM does not guarantee that the product can continue to run successfully, or without error.

On a Windows platform, remove the WebSphere MQ tray icon if it present. The WebSphere MQ tray icon in the lower right corner indicates that a WebSphere MQ process (**amqmtbrn.exe**) is running. Right click the tray icon and click **Hide** to remove it.

Command name

updateWizard.sh and updateWizard.bat, graphical interface to the update installer (installer.jar) file

Related command

updateSilent.sh and updateSilent.bat, command-line interface to the update installer (installer.jar) file

Starting the wizard

The **updateWizard** command (*install_root/update/updateWizard.sh* and *install_root\update\updateWizard.bat*) launches the wizard interface to the update installer application.

For example:

```
install_root/update/updateWizard.sh
install_root\update\updateWizard.bat (Windows only)
```

Parameters

Apply zero, one, or more of these optional parameters in any order, by issuing the **updateWizard** command from the command line.

-dplInstall

Lets you install an interim fix when there are prerequisite check errors.

The default behavior of the update installer application prevents further action if prerequisites are missing.

-dpUninstall

Lets you uninstall an interim fix when there are prerequisite check errors.

-fixOnly

Allows interim fix installation and uninstallation only.

To install and uninstall interim fixes, the `updateWizard` does not require a local copy of the IBM Developer Kit or the Java Runtime Environment (JRE) in a client environment. This option bypasses making a local copy of the IBM Developer Kit or the JRE.

The default action for the update installer enables both interim fix and fix pack installation and uninstallation. The wizard installs a temporary version of one of the IBM products that WebSphere Application Server uses to support the Java 2 SDK on your operating system platform, such as the IBM Developer Kit for AIX, Java Technology Edition.

The **`updateWizard`** command copies the IBM SDK from the `JAVA_HOME` directory to the directory where you are running the **`updateWizard`** (usually `install_root/update`). If the IBM SDK is already in the directory (for example, if you used the **`updateWizard`** command before), it is not necessary for the **`updateWizard`** to make a new copy.

The copy of the IBM SDK remains in the directory until you remove it. The IBM SDK requires approximately 43 MB of free space.

If you install on a client platform, where there is a Java runtime environment instead of the IBM SDK, the **`updateWizard`** copies the JRE to the `install_root/update` directory. The JRE requires about 18 MB of free space.

-usage

Displays a list of parameters and how to use them in the correct command syntax.

Examples of use

Displaying usage information

This command displays information about command syntax.

```
updateWizard -usage
```

Bypassing the local copy of the IBM Developer Kit

The following command bypasses making a local copy of the IBM Developer Kit. Installing and uninstalling fixes does not require the local copy.

```
updateWizard -fixOnly
```

Disabling prerequisite checking

Disabling prerequisite checking is recommended only as directed by IBM Support personnel. Disabling prerequisite checking can leave the installation in a non-valid state, unless done with caution and guidance.

To disable prerequisite checking when installing interim fixes:

```
updateWizard -fixOnly -dpInstall
```

To disable prerequisite checking when uninstalling interim fixes:

```
updateWizard -fixOnly -dpInstall
```

To disable prerequisite locking when installing and when uninstalling interim fixes:

```
updateWizard -fixOnly -dpInstall -dpUninstall
```

Panel descriptions

Panels in the wizard let you select installable fixes and fix packs, view installed fixes and fix packs, and view prerequisite fixes:

- **General**
 - Welcome panel

- Product selection panel
- Menu panel
- **Interim fix installation and uninstallation**
 - Fix repository specifier panel
 - Installable fix selection panel
 - Uninstallable fix selection panel
 - Prerequisite check panel
 - Pre-installation and pre-uninstallation summary panels
 - Installation and uninstallation
 - Post installation and post uninstallation summary panels
- **Fix pack installation and uninstallation**
 - Fix pack repository specifier panel
 - Fix pack selection panel
 - Fix pack features selection panel
 - Pre-installation and pre-uninstallation summary panels
 - Installation and uninstallation
 - Post installation and post uninstallation summary panel

Welcome panel

Use this panel to view a welcome message that contains a brief summary of the update wizard interface, or to link to the Support Web site. (This link is not available on some UNIX-based platforms.) You can also view relevant legal notices.

Product Selection panel

Use this panel to select an installed WebSphere Application Server product. If the wizard cannot detect an installed product, specify the product location in the **Installation directory** field. To make corrections or enter another product location, click **Specify product location**.

On some platforms, there is a limitation in the InstallShield for MultiPlatforms (ISMP) program that the update installer program uses. The ISMP program does not recognize previous installations of WebSphere Application Server on some operating platforms. To work around the problem, click **Specify product location** and type the fully qualified installation root directory for the existing product in the **Installation directory** field.

Menu panel

Use this panel to install or uninstall interim fixes, or to install or uninstall fix packs. If you started the wizard in -fixOnly mode, fix pack options are disabled.

Interim Fix Repository Specifier panel

Use this panel to provide the interim fix repository location in a directory input field. Specify the directory location of the downloaded fix JAR files. The default location for the repository is the *install_root/update/fixes* directory.

Installable Interim Fix Selection panel

Use this panel to select one or more installable fixes for installing. Installed fixes do not appear in the list. Only uninstalled fixes or partially-installed fixes appear. The list includes fix ID name, build date, and the current applied state (uninstalled or partially-installed). Click **Details** for detailed information about selected fixes. The window that appears contains build version information, a long description, and a list of installation prerequisites.

About installation status

An interim fix or fix pack is a collection of updates to one or more product components. Depending on installed product components and on update installer selections you make, the update installer applies either a full or partial set of interim fix or fix pack updates to product components.

Installed status implies that the interim fix or fix pack has no more updates to product components that you can install.

Partially installed status implies that you have updated one or more product components, but the interim fix or fix pack has at least one more update to apply to an installed product component. (There might be other updates to product components you never installed. These updates do not count in the status determination.)

Uninstalled status implies that you have not updated a single product component.

Examples of partially installed states: Several scenarios can lead to a partial installation of an interim fix or fix pack:

- Installation fails, leaving some component updates applied and some unapplied. This is a partial installation accompanied by error messages that describe the problem.
- Skipping certain optional component updates, such as might be present in a fix pack for these features: IBM HTTP Server or embedded messaging. This is a partial installation.
- Dynamically changing interim fix or fix pack status because:
 1. You apply all changes in an interim fix or fix pack that results in an installed status.
 2. You reinstall the WebSphere Application Server product to select an additional, optional feature.
 3. The interim fix or fix pack contains an update to a product component you just installed.

The status changes dynamically from installed to partially installed.

Uninstallable Fix Selection panel

Use this panel to select one or more installed interim fixes for uninstalling. Available interim fixes do not appear in the list. Only installed interim fixes appear. The list includes fix ID name, build date, and the current applied state (installed). Click **Details** to obtain more detailed information about a selected interim fix. The window that appears contains build version information, a long description, and a list of installation prerequisites.

You must uninstall all interim fixes before uninstalling the base WebSphere Application Server product, the Network Deployment product, or the Enterprise product.

Prerequisite Check panel

Use this panel to view prerequisite information when a selected fix has prerequisite fixes that are not installed. You cannot click **Next** until you correct the problem. The `-dpInstall` and the `-dpUninstall` parameters can override this lock, to let you continue with the installation or uninstallation despite prerequisite failure.

Pre-installation Summary and Pre-uninstallation Summary panels

Pre-installation Summary panel Use this panel to display a summary of interim fixes that are selected for installation, the WebSphere Application Server product each interim fix is for, and the directory where each interim fix is located.

Pre-uninstallation Summary panel Use this panel to display a summary of interim fixes that are selected for uninstallation, and the WebSphere Application Server product to which each interim fix is currently applied.

Installation and Uninstallation panel

Installation action Use this panel to view the progress of installing selected fixes. Click **cancel** to revert the installation. Once cancelled, a message confirms that installed fixes are being rolled back. A similar progress panel then appears, to monitor the progress of rolling back the installation of selected fixes.

Uninstallation action Use this panel to view the progress of uninstalling selected fixes.

Post Installation Summary and Post Uninstallation Summary panels

Post Installation Summary panel Use this panel to view the results of the installation. Depending on the result, this panel can display a success message, a failure message, or a canceled message. When the success message appears, the installation process is complete. Click **Finish** to exit the panel. You can also go back and install additional interim fixes, which takes you to the **Menu** panel.

Post Uninstallation Summary panel Use this panel to view the results of the uninstallation. Depending on the result, this panel can display a success message, a failure message, or a canceled message. When the success message appears, the uninstallation process is complete. Click **Finish** to exit the panel. You can go back and uninstall additional interim fixes, which takes you to the **Menu** panel.

Fix Pack Repository Specifier panel

Use this panel to provide the fix pack repository location in a directory input field. The location should point to the directory where you unpacked downloaded fix pack JAR files. The default location for the repository is the `install_root/update/fixpacks` directory.

Fix Pack Selection panel

Use this panel to select from a list of installable fix packs. The panel displays fix packs by ID name, with a radio button next to each for selecting a single fix pack. Also displayed is the build date and current applied state (uninstalled or partially-installed) for each fix pack.

A fix pack on this panel can be in a partially installed state. No installed fix packs appear in the list. Click **Details** for more information about a selected fix pack. The window that appears contains build version information, a long description, installation prerequisites, and a list of included fixes.

See Installable Interim Fix Selection panel for a description of installation states.

Fix Pack Features Selection panel

Use this panel to view a list of WebSphere Application Server features with optionally installable service in the selected fix pack. Features that can appear include IBM HTTP Server and the embedded messaging feature, which is based on IBM WebSphere MQ technology.

If you do not install optional service for an installed feature, the fix pack installs successfully as a **partially installed fix pack** because there is service that you did not install.

If you installed the IBM HTTP Server product as a feature, use the update installer to update it with service in an interim fix or fix pack. Otherwise, download an updated IBM HTTP Server product and install it into the same directory as your existing version, to update the existing installation. You can also uninstall the current version and install the downloaded version to avoid any issues with migration.

Update your configuration if you reinstall. The process is described in the *Manually configuring supported Web servers* (tins_manualWebserver) topic in the base Application Server InfoCenter.

Always apply any outstanding corrective service to the stand-alone IBM WebSphere MQ product if you have it, before using the WebSphere Application Server update installer to update the embedded messaging feature with service in an interim fix or fix pack.

Do not install service to the embedded messaging feature if you install corrective service to the stand-alone IBM WebSphere MQ product.

Pre-installation Summary and Pre-uninstallation Summary panels

Pre-installation Summary panel Use this panel to display a summary of the fix pack selected for installation, the WebSphere Application Server product the fix pack is for, and the directory where the fix pack is located.

Pre-uninstallation Summary panel Use this panel to display a summary of the fix pack selected for uninstallation, the WebSphere Application Server product to which the fix pack was applied, and the directory where the fix pack is located.

Installation and Uninstallation panels

Installation action Use this panel to view the progress of installing the selected fix pack. Click **cancel** to revert the installation. Once cancelled, a message confirms that the fix pack is being rolled back. A similar progress panel then appears, to monitor the progress of rolling back the installation of the selected fix pack.

Uninstallation action Use this panel to view the progress of uninstalling the selected fix pack. There is no way to cancel the uninstallation.

You must uninstall all fix packs before uninstalling the base WebSphere Application Server product, the Network Deployment product, or the Enterprise product.

Post Installation Summary and Post Uninstallation Summary panel

Post Installation Summary panel Use this panel to view the results of the installation. Depending on the result, this panel can display a success message, a failure message, or a canceled message. When the success message appears, the installation process is complete. Click **Finish** to exit the panel.

You can go back and install another fix pack, which takes you to the **Menu** panel.

Post Uninstallation Summary panel Use this panel to view the results of the uninstallation. Depending on the result, this panel can display a success message, a failure message, or a canceled message. When the success message appears, the uninstallation process is complete. Click **Finish** to exit the panel.

You can go back and uninstall another fix pack, which takes you to the **Menu** panel.

Chapter 5. Product version and history information

The WebSphere Application Server product contains structural differences from previous versions. The `/properties/version` directory in the installation root contains important data about the product and its installed components, such as the build version and build date. This information is included in `[product].product` and `[component].component` files.

The `/properties/version/history` directory in the installation root contains a collection of records for installed fixes and fix packs. This information is included in `[interim fixID].efixApplied`, `[interim fixID].efixDriver`, `[fix packID].ptfApplied`, and `[fix packID].ptfDriver` files.

A driver file has useful information about the entire contents of an interim fix or fix pack. The applied file has relevant information about the fixes or fix packs that are currently applied.

Event.history files contain a detailed log about updates you have applied, either successfully or unsuccessfully. Time-stamped, detailed logs record each update process in the `/properties/version/log` directory of the installation root. Beginning with Fix Pack 2, the time-stamped, detailed logs are in the `install_root/logs/update` directory.

This topic describes the XML data files that store product information for V5 WebSphere Application Server products. By default, the document type declarations (DTDs) for these files are in the `properties/version/dtd` folder of the installation root, or the server root directory. See the Storage locations section for more information.

This topic includes:

- A list of product information files and file locations
- Report scripts for displaying version and history information
- A description of logs and component backups
- A list of storage locations
- A description of how the update service makes operational use of the product information
- A data dictionary that describes data in the files

Product information files

There are two kinds of product information files:

- XML files in the `/properties/version` directory that store version information
- XML files in the `/properties/version/history` directory

XML files in the `/properties/version` directory that store version information

The following file indicates that a WebSphere Application Server V5 product is installed:

platform.websphere

One file whose existence indicates that a WebSphere Application Server product is installed. An example of the file follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE websphere PUBLIC "websphereId" "websphere.dtd">
<websphere name="IBM WebSphere Application Server" version="5.0"/>
```

The following XML files in the `/properties/version` directory represent installed items and installation events:

<product-id>.product

One file whose existence indicates the particular WebSphere Application Server product that is installed. Data in the file indicates the version, build date, and build level. For example, the file might be the ND.product file, which indicates that the installed product is WebSphere Application Server Network Deployment. An example of the file follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE product PUBLIC "productId" "product.dtd">
<product name="IBM WebSphere Application Server for Network Deployment">
  <id>ND</id>
  <version>5.0.0</version>
  <build-info date="10/5/02" level="s0239.28"/>
</product>
```

<component-name>.component

Any number of component files that each indicate the presence of an installed component, which is part of the product. Data in the file indicates the component build date, build version, component name, and product version. For example, the file might be the activity.component file, which indicates that the **activity** component is installed. The activity component is part of the Network Deployment product. An example of the file follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE component PUBLIC "componentId" "component.dtd">
<component build-date="10/5/02" build-version="s0239.28" name="activity" spec-version="5.0"/>
```

<extension.id>.extension

Any number of extension files that each indicate the presence of an extension that you install as a user extension, as part of a service engagement, or as installed by a third party product. The <extension.id>.extension files are not created, logged, or removed by WebSphere Application Server products.

<fix-id>.efix

Any number of fix files that each indicate the presence of an installed interim fix.

<ptf-id>.ptf

Any number of files, that each indicate the presence of an installed fix pack.

XML files in the /properties/version/history directory

This file stores version history information:

event.history

One file that lists update events that have occurred. An update event is an operation that installs or uninstalls an interim fix or a fix pack. The file is sorted by the date and time of the events that are listed.

The following XML files in the /properties/version/history directory describe fixes and fix packs that are currently installed. These XML files are related to installation items by the primary ID information, which is shown here by *<angle brackets>* and italicized text.

<fix-id>.efixDriver

Fix-driver defining information

<fix-id>.efixApplied

Fix installation details

<ptf-id>.ptfDriver

fix pack-driver defining information

<ptf-id>.ptfApplied

fix pack installation details

Reports

You can view product information by examining files in the `install_root/properties/version` directory, including the `install_root/properties/version/history` directory.

WebSphere Application Server provides the ability to generate two types of reports about the data in the files, **Version** reports and **History** reports. The following report-generation scripts are available in the installation root `bin` directory.

Product version reports

The following report generation scripts extract data from XML data files in the `properties/version` folder.

versionInfo.bat

Lets you use parameters to create a version report on Windows platforms.

Do not use the **versionInfo** command while installing or uninstalling the product, or while installing or uninstalling an interim fix or fix pack.

Parameters:

-format

TEXT | HTML

Selects the format of the report. The default is TEXT.

-file <fileName>

Specifies the output file name. The report goes to standard output (stdout) by default.

-components

Adds a list of installed components to the report.

-componentDetail

Adds details about installed components to the report.

-efixes

Adds a list of applied interim fixes to the report.

-efixDetail

Adds details about applied interim fixes to the report.

-ptfs Adds a list of applied fix packs to the report.

-ptfDetail

Adds details about applied fix packs to the report.

versionInfo.sh

Lets you use parameters to create a version report on UNIX-based platforms. Parameters are the same as for the Windows version.

Do not use the **versionInfo** command while installing or uninstalling the product, or while installing or uninstalling an interim fix or fix pack.

genVersionReport.bat

Generates the `versionReport.html` report file in the `bin` directory on Windows platforms. The report includes the list of components, interim fixes, and fix packs.

genVersionReport.sh

Generates the `versionReport.html` report file in the `bin` directory on UNIX-based platforms. The report includes the list of components, interim fixes, and fix packs.

Product history reports

The following report generation scripts extract data from XML data files in the `install_root/properties/version/history` folder:

historyInfo.bat

Lets you use parameters to create a history report of installed and uninstalled interim fixes and fix packs, on Windows platforms. Specify a component name to create a report that shows the history for that component.

Parameters:

-format

TEXT | HTML

Selects the format of the report. The default is TEXT.

-file <fileName>

Specifies the output file name. The report goes to standard output (stdout) by default.

-updateID <ID>

Specifies the ID of an fix or fix pack update. When specified, the product history report displays events for only the specified update. When not specified, the report displays events for all updates.

-component <componentName>

Specifies the name of a component. When specified, the product history report displays events for only the named component. When not specified, the report displays events for all components.

historyInfo.sh

Lets you use parameters to create a history report on UNIX-based platforms. Parameters are the same as for the Windows version.

genHistoryReport.bat

Generates the `historyReport.html` report file in the `install_root\bin` directory on Windows platforms. The report includes all updates and components.

genHistoryReport.sh

Generates the `historyReport.html` report file in the `bin` directory on UNIX-based platforms. The report includes all updates and components.

Logs and component backups

WebSphere Application Server products use two other directories when performing update operations, for logging and backups. By default, the two directories are relative to the product version directory, as follows:

install_root/properties/version/log

Product updates log directory from V5.0.0 and V5.0.1.

WebSphere Application Server products store log files to document component, interim fix and fix pack operations and updates.

install_root/logs/update

Product updates log directory beginning with Fix Pack 2, which is Version 5.0.2.

Beginning with Fix Pack 2, the update installer stores its detailed log files in the `install_root/logs/update` directory.

install_root/properties/version/backup

Product updates backup directory

WebSphere Application Server products back up components before applying fixes and fix packs. If you uninstall an interim fix or fix pack, WebSphere Application Server products restore the backed-up component JAR file.

File naming convention

Time stamp

YYYYMMDD_HHMMSS

For example: 20020924_211832 is 24-Sep-2002, 9:18:32 pm, GMT. All time stamps are in GMT.

ID

Fix ID or fix pack ID

For example: apar6789c is an fix ID; PTF_1 is a fix pack ID.

Operation

install | uninstall

Interim fix log file names

<timeStamp>_<fixId>_<operation>.log

For example: properties/version/log/20020924_211832_apar6789c_install.log and properties/version/log/20020924_211912_apar6789c_uninstall.log

At Fix Pack 2 (V5.0.2) or later, the update installer creates these logs:

install_root/logs/update/20020924_211832_apar6789c_install.log and
install_root/logs/update/20020924_211912_apar6789c_uninstall.log

Interim fix component log file names

<timeStamp>_<fixId>_<componentName>_<operation>.log

For example: properties/version/log/20020924_211832_apar6789c_ras_install.log and properties/version/log/20020924_211912_apar6789c_ras_uninstall.log

At Fix Pack 2 or later, the update installer creates these logs:

install_root/logs/update/20020924_211832_apar6789c_ras_install.log and
install_root/logs/update/20020924_211912_apar6789c_ras_uninstall.log

Fix pack log file names

<timeStamp>_<ptfId>_<operation>.log

For example: properties/version/log/20020924_211832_PTF_1_install.log and properties/version/log/20020924_211912_PTF_2_uninstall.log

At Fix Pack 2 or later, the update installer creates these logs:

install_root/logs/update/20030324_211832_was50_fp2_install.log and
install_root/logs/update/20030325_211912_was50_fp2_uninstall.log

Fix pack component log file names

<timeStamp>_<ptfId>_<componentName>_<operation>.log

For example: properties/version/log/20020924_211832_PTF_1_ras_install.log and properties/version/log/20020924_211912_PTF_2_ras_uninstall.log

At Fix Pack 2 or later, the update installer creates these logs:

install_root/logs/update/20030324_211832_was50_fp2_ras_install.log and
install_root/logs/update/20030325_211912_was50_fp2_ras_uninstall.log

Backup JAR file names

<timeStamp>_<ptfId>_<componentName>_undo.jar or

<timeStamp>_<fixId>_<componentName>_undo.jar

For example: 20020924_211832_apar6789c_ras_undo.jar

Do not delete a backup Java archive (JAR) file. You cannot remove a component update if the corresponding backup JAR file is not present.

Update processing might also use a temporary directory, if necessary. A Java property specifies this directory as described in the next section.

Storage locations

Product information files are located relative to the WebSphere Application Server product installation root, or the server root directory.

Default file paths and Java properties that set them are:

Version directory

`install_root/properties/version` or `server_root/properties/version`

History directory

`install_root/properties/version/history`

Updates log directory

`install_root/properties/version/log`

The version of the update installer that is bundled with Fix Pack 2 and later fix packs, stores log files in the `install_root/logs/update` directory.

Updates backup directory

`install_root/properties/version/backup`

DTD directory

`install_root/properties/version/dtd`

Temporary directory

Specified by the `java.io.tmpdir` Java system property

Operational description

WebSphere Application Server products update the product version history information while performing events that install or uninstall fixes or fix packs. Events that might occur include:

- A WebSphere Application Server product adds an interim fix file (with an extension of `.efix`) to the version directory to indicate that an interim fix is currently installed.
- A WebSphere Application Server product removes an interim fix file from the version directory when it uninstalls the corresponding interim fix.
- A WebSphere Application Server product adds an interim fix driver file (with an extension of `.efixDriver`) to the history directory when an interim fix is installed. An interim fix driver file contains defining information for an interim fix.
- A WebSphere Application Server product removes an interim fix driver file when it removes the corresponding interim fix.
- A WebSphere Application Server product adds an interim fix application file (with an extension of `.efixApplied`) to the history directory when it installs an interim fix. An interim fix application file contains information that identifies component updates that have been applied for an interim fix. The application file also provides links to component log and backup files.
- A WebSphere Application Server product removes an interim fix application file when it removes the corresponding interim fix.
- A WebSphere Application Server product adds a fix pack, with an extension of `.ptf`, to the version directory to indicate that a fix pack is currently installed.
- A WebSphere Application Server product removes a fix pack file from the version directory when it uninstalls the corresponding fix pack.
- A WebSphere Application Server product adds a fix pack driver file (with an extension of `.ptfDriver`) to the history directory when it installs a fix pack. A fix pack driver file contains defining information for a fix pack.

- A WebSphere Application Server product adds a fix pack application file (with an extension of .ptfApplied) to the history directory when it installs a fix pack. A fix pack application file contains information that identifies component updates that have been applied for a fix pack. The application file also provides links to component log and backup files.
- A WebSphere Application Server product makes entries in the history file, event.history, when it installs or uninstalls an update.
- A WebSphere Application Server product stores a parent event for each fix that it installs or uninstalls.
- A WebSphere Application Server product stores a parent event for each fix pack that it installs or uninstalls.
- A WebSphere Application Server product stores child component events for each component update that it installs or uninstalls, beneath the corresponding fix or fix pack parent event.
- A WebSphere Application Server product stores one log file in the log directory as it installs or uninstalls one fix or fix pack.
- A WebSphere Application Server product stores one log file in the log directory as it installs or uninstalls an interim fix or fix pack, in response to each component update that occurs.
- A WebSphere Application Server product stores a component backup file in the backup directory for each component update that it installs.
- A WebSphere Application Server product removes a component backup file from the backup directory for each component update that it uninstalls.

Data dictionary

Type Family: **websphere product family**

File Types: websphere

File Type: websphere

Elements:	name	string	required
	version	string	required

Persistence: <versionDir>/platform.websphere

Type Detail:

The websphere file is placed to denote the presence of websphere family products.

Element Detail:

websphere.name	The WebSphere product family name.
websphere.version	The WebSphere product family version.

Type Family: product

File Types: product
 component
 extension

File Type: product

Persistence: <versionDir>/<id>.product

Elements:	id	string	required
	name	string	required
	version	string	required
	build-info	complex	required

Type Detail:

A product file is placed to denote the presence of a specific WebSphere family product. The product's id is embedded in the product file name.

Element Detail:

product.id	The id of the product.
product.name	The name of the product.
product.version	The version of the product.
product.build-info	An element containing build information for the product.

Element Type: build-info

Elements:	date	date	required
	level	string	required

Type Detail:

A build-info instance details the build of a specific installed websphere family product.

Element Detail:

build-info.date	The date on which the product was build.
build-info.level	The level code of the product's build.

File Type: component

Persistence: <versionDir>/<name>.component

Elements:	name	string	required
	spec-version	string	required
	build-version	string	required
	build-date	date	required

File Detail:

A component file denotes the presence of a specific component. The component name is embedded in the component file name.

Element Detail:

component.name	The name of the component.
component.spec-version	The specification version of the component.
component.build-version	The build level of the component.
component.build-date	The build date of the component.

File Type: extension

Persistence: <versionDir>/<id>.extension

Elements:	id	string	required
	name	string	required

File Detail:

An extension file denotes the presence of a specific extension. The extension's id is embedded in the extension file name.

The elements of an extension file are minimally specified. The listed elements are required. Additional elements may be present as determined by the actual installed extension.

Element Detail:

extension.id	The id of the extension.
--------------	--------------------------

extension.name The name of the extension.

Type Family: update

File Types: efix
 ptf
 efix-applied
 ptf-applied

File Type: efix

Persistence: <versionDir>/<id>.efix

Elements:	id	string	required
	apar-number	string	optional
	pmr-number	string	optional
	short-description	string	required
	long-description	string	required
	is-temporary	boolean	required
	build-version	string	required
	build-date	date	required
	component-update	complex	min=1, max=unbounded
	platform-prereq	complex	min=0, max=unbounded
	product-prereq	complex	min=0, max=unbounded
	efix-prereq	complex	min=0, max=unbounded
	custom-property	complex	min=0, max=unbounded

Type Detail:

An efix file denotes the presence of some portion of a specific interim fix. The id of the fix is embedded in the file name.

An efix file contains all fix data, such as description, a listing of component updates, and prerequisite information.

Almost always, when installing an interim fix, all of the potential component updates within the fix are required to be installed.

A separate application file must be examined to determine the components which have been updated for a particular interim fix.

A list of custom properties may be provided. These are provided for future use.

Element Detail:

efix.id	The id of the interim fix.
efix.short-description	A short description of the interim fix.
efix.long-description	A long description of the interim fix.
efix.is-trial	A flag indicating whether or not an interim fix is considered a trial interim fix. Generally, a trial fix will be followed up with a more permanent interim fix.
efix.expiration-date	A date on which the fix is to be considered obsolete.
efix.build-version	The build version of the interim fix. This is distinct from the build version of component updates contained within the interim fix.
efix-build-date	The build date of the interim fix. This is distinct from the build version of the component updates contained within the interim fix.
efix.apar-info	A list of APAR's which are associated with the interim fix.

efix.component-update	A list of updates for components. For an interim fix, these are usually all required, and are all patch updates. At least one component update must be present.
efix.efix-prereq	A list of prerequisite fixes for the interim fix. Note that prerequisite fixes may be negative (see below). This list may be (and is often) empty.
efix.plaform-prereq	A list of platforms on which the fix may be installed. This list may be empty, in which case the fix may be installed on all platforms.
efix.product-prereq	A list of products on which the fix may be installed. This list may be empty, in which case the fix may be installed on all products.
efix.custom-proppty	A list of properties, provided for future use.

File Type: ptf

Persistence: <versionDir>/<id>.ptf

Elements:	id	string	required
	short-description	string	required
	long-description	string	required
	build-version	string	required
	build-date	date	required
	component-update	complex	min=1, max=unbounded
	product-update	complex	min=0, max=unbounded
	platform-prereq	complex	min=0, max=unbounded
	product-prereq	complex	min=0, max=unbounded
	included-efix	complex	min=0, max=unbounded
	custom-property	complex	min=0, max=unbounded

Type Detail:

A ptf file denotes the presence of some portion of a specific fix pack. The id of the fix pack is embedded in the fix pack file name.

A ptf file contains all fix pack data, such as description, a listing of component updates, and prerequisite information.

Usually, when installing a fix pack, omit certain potential component updates, but only when the corresponding component is not installed.

Examine a separate application file to determine which components a particular fix pack has updated.

A fix pack can include updates for a number of fixes.

A list of custom properties might be provided. These are provided for future use.

Element Detail:

ptf.id	The ID of the fix pack.
ptf.short-description	A short description of the fix pack.
ptf.long-description	A long description of the fix pack.
ptf.build-version	The build version of the fix pack. This is distinct from the build version of component updates contained within the fix pack.
ptf-build-date	The build date of the fix pack. This is distinct from the build version of the component updates contained within the fix pack.

ptf.component-update	A list of updates for components. For a fix pack, these are usually all required, and are all patch updates. At least one component update must be present.
ptf.plaform-prereq	A list of platforms on which you can install the fix pack. This list might be empty. If so, you can install the fix pack on all platforms.
ptf.product-prereq	A list of products on which you can install the fix pack. This list might be empty. If so, you can install the fix pack on all products.
ptf.included-efix	A list of fixes which are included (fixed) by the fix pack.
ptf.custom-property	A list of properties, provided for future use.

Element Type: component-update

Elements:	component-name	string	required
	update-type	enum	required [enumUpdateType]
	is-required	boolean	required
	is-recomended	boolean	required
	is-optional	boolean	required
	is-external	boolean	required
	root-property-file	anyURL	optional
	root-property-name	string	optional
	root-property-value	anyURL	optional
	is-custom	boolean	required
	primary-content	string	required
	component-prereq	complex	min=0, max=unbounded
	final-version	complex	optional
	custom-property	complex	min=0, max=unbounded

Type Detail:

A component update represents a potential component update which is packaged in an update (an interim fix or a fix pack).

An component update may be required, in which case the parent update may not be installed unless the component update can be installed. (A component update can be installed if the corresponding component is installed.)

A component update may be a custom update, in which case the content which was provided must be an executable file. Otherwise, the content which is provided must be an update jar file.

A component update has a type. A final version may be required according to the update type.

Element Detail:

component-update.component-name	The name the component which is to be updated.
component-update.update-type	The type of the component update, one of 'add', 'replace', 'remove', or 'patch'. Final version information must be provided when the update type is 'add' or 'replace'.
component-update.is-required	A flag which, when true, specifies that the parent update may not be applied unless this component update is applied.
component-update.is-recomended	A flag which, when true, specifies that this component update, although optional, should be installed.
component-update.is-optional	A flag which, when true, specifies that this update may be omitted even if its corresponding component

is installed.

component-update.is-external	A flag which, when true, specifies that this component update may live outside of the usual install root.
component-update.root-property-file	For a component with an external root, this properties file provides the root value.
component-update.root-property-name	For a component with an external root, this named property provides the root value.
component-update.root-property-value	For a component with an external root, this value provides the default root value.
component-update.is-custom	A flag which, when true, specifies that the update is a custom update. When true, the content must be an executable program. When false, the content must be an update jar.
component-update.primary-content	The name of the content which is provided for the update. This will be an entry which is packaged in the 'components' directory of the update.
component-update.component-prereq	A list of component versions, one of which must be present for this update to be installed. When this list is empty, any component version is allowed.
component-update.final-version	Final version information for the component. A final version is required when the update operation is 'add' or 'replace'.
component-update.custom-property	A list of properties, provided for future use.

Element Type: apar-info

Elements:	number	string	required
	date	date	required
	short-description	string	required
	long-description	string	optional

Type Detail:

An apar-info object provides information about an APAR which is associated with an interim fix, usually indicating that the fix provides an interim fix for the APAR.

Element Detail:

apar-info.number	The number of the associated APAR.
apar-info.date	The date of the APAR.
apar-info.short-description	A short description of the APAR.
apar-info.long-description	An optional long description of the APAR.

Element Type: efix-prereq

Elements:	efix-id	string	required
	is-negative	boolean	required
	install-index	int	optional

Type Detail:

An interim fix prerequisite instance denotes an interim fix that must be present (or, if negative, must be absent) for the parent fix to be installed.

efix prerequisite instances may specify a cycle, in which case the prerequisite specification is treated as a corequisite specification.

The following chart summarizes the interpretation of prerequisite information for two fixes:

fix1	fix2	
-	-	The fixes may be installed without regard to each other.
fix2	-	fix1 must be installed after fix2 is installed.
-	fix1	fix2 must be installed after fix1 is installed.
fix2	fix1	fix1 and fix2 must be installed together.
!fix2	-	fix1 may not be installed after fix2 is installed.
-	!fix1	fix2 may not be installed after fix1 is installed.
!fix2	!fix1	fix1 and fix2 may not ever be installed together.
!fix2	fix1	This is an erroneous specification.
fix2	!fix1	This is an erroneous specification.

The installation index element provides ordering information for corequisite fixes that must be installed in a particular order.

Element Detail:

fix-prereq.efix-id	The id of the prerequisite interim fix.
fix-prereq.is-negative	A flag which indicates if the prerequisite fix is required or prohibited. If false, install the interim fix before installing the parent interim fix. If true, do not install the interim fix before you install the parent interim fix.
fix-prereq.install-index	An optional index number used to order corequisite fixes.

Element Type: product-update

Elements:	product-id	string	required
	product-name	string	required
	build-version	string	required
	build-date	date	required
	build-level	string	required

Type Detail:

A product update specifies a replacement to a product file.

The product update information matches the information in product files.

Multiple product updates may be present, in which case each matching product is updated.

Element Detail:

product-update.product-id	The id of the product that is updated.
product-update.product-name	The name of the product.
product-update.build-version	The build version of the product.
product-update.build-date	The build date of the product.
product-update.build-level	The build level of the product.

Element Type: component-prereq

Elements:	component-name	string	required
	spec-version	string	required
	build-version	string	required
	build-date	date	required

Element Type: platform-prereq

Elements:	architecture	string	required
	os-platform	string	optional

os-version	string	optional
------------	--------	----------

Type Detail:

A platform prerequisite instance denotes a platform which must be present for an update to be installed. The element values are according to the values supplied for the matching java properties.

Note that when multiple platform prerequisites are specified, these prerequisites have an OR relationship: At least one of the platform prerequisites must be satisfied.

Element Detail:

platform-prereq.architecture	The name of an architecture which must be present.
platform-prereq.os-platform	The name of an operating system which must be present. This element is optional. When absent, the architecture is checked, but the os-platform and os-version are not.
platform-prereq.os-version	The version of a the operating system which must be present. This element is optional. When absent, the architecture and os-platform are checked, but os-version is not. (When os-platform is absent, os-version should not be set.)

Element Type: product-prereq

Elements:	product-id	string	required
	build-version	string	optional
	build-date	date	optional
	build-level	string	optional

Type Detail:

A product prerequisite specifies that a particular product must be present for an update to be installed.

Note that when multiple product prerequisites are specified, these prerequisites have an OR relationship: At least one of the product prerequisites must be satisfied.

Note that all of the elements are required. When multiple products having the same id are supported by an update, multiple product prerequisites must be specified.

Element Detail:

product-prereq.product-id	The id of the product which must be present.
product-prereq.build-version	The version of the product which must be present.
product-prereq.build-date	The build date of the product which must be present.
product-prereq.build-level	The level date of the product which must be present.

Element Type: component-prereq

Elements:	component-name	string	required
	spec-version	string	required
	build-version	string	required
	build-date	date	required

Type Detail:

A version prerequisite specifies that a particular component version must be present for an update to be installed.

Note that when multiple version prerequisites are specified, these prerequisites have an OR relationship: At least one of the version prerequisites must be satisfied.

Element Detail:

version-prereq.component-name	The name of the component which must be present.
version-prereq.spec-version	The specification version of the component which must be present.
version-prereq.build-version	The version of the component which must be present.
version-prereq.build-date	The build date of the component which must be present.

Element Type: included-efix

Elements:	efix-id	string	required
-----------	---------	--------	----------

Type Detail:

An included-efix identifies an interim fix by ID and indicates that the fix is included in the fix pack.

Element Detail:

included-efix.efix-id	The ID of the fix that the fix pack includes.
-----------------------	---

Element Type: custom-property

Elements:	property-name	string	required
	property-type	string	optional
	property-value	string	optional

Type Detail:

A custom property encodes a key-value pair, with an optional type element. Custom properties are provided for future use.

Element Detail:

custom-property.property-name	The name of the custom property.
custom-property.property-type	An optional type of the custom property. The semantics of this type are defined by user of the property value.
custom-property.property-value	The value of the custom property.

File Type: efix-applied

Persistence: <versionDir>/<id>.efixApplied

Elements:	efix-id	string	required
	component-applied	complex	min=0, max=unbounded

Type Detail:

An efix-applied collection specifies what components have been updated for the fix as specified by the efix id.

Element Detail:

efix-applied.efix-id	The id of the fix for which applieds are recorded.
efix-applied.component-applied	The list of recorded applications.

File Type: ptf-applied

Persistence: <versionDir>/<id>.ptfApplied

Elements: ptf-id string required
 component-applied complex min=0, max=unbounded

Type Detail:

A ptf-applied collection specified what components have been updated for the fix pack as specified by the fix pack ID.

Element Detail:

ptf-applied.efix-id The ID of the fix pack for which applieds are recorded.

ptf-applied.component-applied The list of recorded applications.

Element Type: component-applied

Elements: component-name string required
 update-type enum required [enumUpdateType]
 is-required boolean required
 is-optional boolean required
 is-external boolean required
 root-property-file anyURL optional
 root-property-name string optional
 root-property-value string optional
 is-custom boolean required
 log-name anyURL required
 backup-name anyURL required
 time-stamp date required
 initial-version complex optional
 final-version complex optional

Type Detail:

An applied instance is present to indicate the application of an update for a particular fix or fix pack to a particular component. (The particular fix or fix pack is as specified by the applied's parent.) An applied provides sufficient information to undo itself.

The elements of an applied are copies of values from update events.

Element Detail:

component-applied.component-name The name of the component which was updated.

component-applied.update-type The type of the component update.

component-applied.is-required A flag which, when true, specifies that the parent update requires this component update.

component-applied.is-optional A flag which, when true, specifies that the parent update does not require this component update, even if the component is installed.

component-applied.is-external A flag which, when true, specifies that this component update was applied to a location different than the usual install_root.

component-applied.root-property-file For an update against a component having an external root, this properties file provides the root value.

component-applied.root-property-name For an update against a component having an external root, this named property provides the root value.

component-applied.root-property-value For an update against a component having an external root, this is a record of the

actual external root.

component-applied.is-custom	A flag which, when true, specifies that the application was a custom update. When true, an executable program was applied. When false, the contents of an update jar were applied.
component-applied.log-name	The name of the log file which was generated by this application.
component-applied.backup-name	The name of the backup file which was generated by this application.
component-applied.time-stamp	The time of this application (the ending time of the corresponding update event).
component-applied.initial-version	The version of the component before the application. This version will be null if the application was an add.
component-applied.final-version	The version of the component after the application. This will be null if the update was a removal.

Element Type: initial-version

Elements:	component-name	string	required
	spec-version	string	required
	build-version	string	required
	build-date	string	required

Type Detail:

A initial-version instance is used to describe a component level as the initial version of a component.

Element Detail:

initial-version.component-name	The name of the component.
initial-version.spec-version	The new specification version for the component following the update.
initial-version.build-version	The new build version for the component.
initial-version.build-date	The new build date for the component.

Element Type: final-version

Elements:	component-name	string	required
	spec-version	string	required
	build-version	string	required
	build-date	string	required

Type Detail:

A final-version instance is used to supply a component level for a component which has been added or replaced.

Element Detail:

final-version.component-name	The name of the new component.
final-version.spec-version	The new specification version for the component following the update.
final-version.build-version	The new build version for the component.
final-version.build-date	The new build date for the component.

Enum Type: enumUpdateType

Values:	0 add
	1 replace
	2 remove
	3 patch

Type Detail:

An update type instance specifies the type of an update. An 'add' update adds

a component into an installation. A 'replace' update replaces a particular version of a component with a different version of that component. A 'remove' update removes a component. A 'patch' update performs a limited update to a component, in particular, without changing the version of the component.

When adding a component, that component may not already be present.

When replacing or removing a component, that component must be present.

When patching a component, that component must be present.

When replacing or removing a component, or when patching a component, usually, at least one version prerequisite will be specified for the component update.

Value Detail:

enumUpdateType.add	Specifies that an update adds a component.
enumUpdateType.replace	Specifies that an update replaces a component.
enumUpdateType.remove	Specifies that an update removes a component.
enumUpdateType.patch	Specifies that an update modifies a component, but does not change its version.

Type Family: history

File Type: event-history

Persistence: <historyDir>/event.history

Elements: update-event complex min=0, max=unbounded

Type Detail:

One event history is provided for a websphere product family installation. This event history contains history of update events, corresponding with the actual update events for that product family.

Element Detail:

event-history.update-event	The list of update events for the websphere product family. The top level events are fix and fix pack events, each containing one or more component events.
----------------------------	---

Element Type: update-event

Elements:	event-type	enum	required [enumEventType]
	parent-id	string	required
	id	string	required
	update-type	enum	required [enumUpdateType]
	is-required	boolean	required
	is-optional	boolean	required
	is-external	boolean	required
	root-property-file	anyURL	optional
	root-property-name	string	optional
	root-property-value	string	optional
	is-custom	boolean	required
	primary-content	anyURI	required
	event-action	enum	required [enumEventAction]
	log-name	anyURI	required
	backup-name	anyURI	required
	start-time-stamp	dateTime	required
	end-time-stamp	dateTime	optional

status	enum	optional [enumEventResult]
status-message	string	optional
initial-version	complex	optional
final-version	complex	optional
update-event	complex	optional

Type Detail:

An update event denotes a single update action, applying to either a fix, a fix pack, or to a component, according to the set event type.

Fix (efix) and fix pack (ptf) type events each have a collection of component events.

Currently, component events have no child events.

Element Detail:

update-event.event-type	The type of this event, either an interim fix or fix pack (ptf) type event, or a component type event.
update-event.parent-id	This element is present only for component events. The ID of the parent fix or fix pack of this event.
update-event.id	The ID of the fix, fix pack, or component that was updated, interpreted according to the type of the event.
update-event.update-type	The type of update for component events.
update-event.is-required	A flag which, when true, specifies that this component update is required.
update-event.is-optional	A flag which, when true, specifies that this component update is optional, even if the component is installed.
update-event.is-external	A flag which, when true, specifies that this update used an external root.
update-event.root-property-file	For an update of an external component, this properties file contains the external root value.
update-event.root-property-name	For an update of an external component, the property having this name specifies the external root value.
update-event.root-property-value	For an update of an external component, the root value.
update-event.is-custom	A flag that, when true, specifies that the application was a custom update. When true, an executable program was applied. When false, the contents of an update jar were applied.
update-event.primary-content	The URL of the primary content for the update.
update-event.event-action	The type of action for this event.
update-event.log-name	The name of the log file which was generated for this event.
update-event.backup-name	The name of the backup file which was generated for this event.
update-event.start-time-stamp	The XML timestamp of the starting time of the

event. This timestamp follows the XML timestamp format, meaning that time zone information is included.

update-event.end-time-stamp	The XML timestamp of the ending time of the event. This timestamp follows the XML timestamp format, meaning that time zone information is included. When absent, the update operation corresponding to the parent event failed with a non-recoverable exception.		
update-event.status	The result of the update.		
update-event.status-message	Message text provided in addition to the basic status code. Exception text is provided through the status-message when an update fails.		
update-event.initial-version	This element is not used unless the update is a component type update. The initial version of the component which was updated. This element is absent when the update is an add type update.		
update-event.final-version	This element is not used unless the update is a component type update. The final version of the component which was updated. This element is absent when the update is a remove type update.		
update-event.update-event	A collection of child events. This collection is used for fix and fix pack type events. This collection is empty for component type events.		

Element Type: initial-version

Elements:	spec-version	string	required
	build-version	string	required
	build-date	string	required

Type Detail:

A initial-version instance is used to describe a component level as the initial version of a component.

Element Detail:

initial-version.spec-version	The new specification version for the component following the update.		
initial-version.build-version	The new build version for the component.		
initial-version.build-date	The new build date for the component.		

Element Type: final-version

Elements:	spec-version	string	required
	build-version	string	required
	build-date	string	required

Type Detail:

A final-version instance is used to supply a component level for a component which has been added or replaced.

Element Detail:

final-version.spec-version	The new specification version for the component following the update.		
----------------------------	---	--	--

final-version.build-version The new build version for the component.

final-version.build-date The new build date for the component.

Enum Type enumEventType

Values: 0 Fix (efix)
1 fix pack (ptf)
2 Component

Type Detail:

An event type instance specifies the type of an update event, which is either an interim fix (efix) event, a fix pack (ptf) event or a component event. The interpretation of particular event elements depends on the set event type.

Value Detail:

enumEventType.efix Specifies that an event is for an interim fix update.

enumEventType.ptf Specifies that an event is for a fix pack update.

enumEventType.component Specifies that an event is for a component update.

Enum Type: enumEventAction

Values: 0 Install
1 Uninstall
2 Selective install
3 Selective uninstall

Type Detail:

An event action instance specified the operation performed by an update, which can be an install or uninstall operation, and which may be a selective operation. Component operations are always either install or uninstall type operations, only fix and fix pack operations may be selective operations.

A selective operation is an installation which is applied to a preset list of components. In particular, potential component updates may be skipped, and component updates which were already applied may be reapplied.

A selective uninstall operation is used to back out an update which was cancelled by the user.

Value Detail:

enumEventAction.install Specifies that an event is an install operation.

enumEventAction.uninstall Specifies that an event is an uninstall operation.

enumEventAction.selective-install Specifies that an event is an install operation with a preset list of components which are updated.

enumEventAction.selective-uninstall Specifies that an event is an install operation with a preset list of components which are updated.

Enum Type: enumUpdateType

Values: 0 Add
1 Replace
2 Remove
3 Patch

Type Detail:

An update type instance specifies the type of a component update. An 'add' update adds a component into an installation. A 'replace' update replaces a particular version of a component with a different version of that component. A 'remove' update removes a component. A 'patch' update performs a limited update to a component, in particular, without changing the version of the component.

To add a new component, the component must not exist. To replace or remove a component, the component must exist. To patch a component, the component must exist.

When replacing or removing a component, or when patching a component, usually, at least one version prerequisite is specified for the component update.

Value Detail:

<code>enumUpdateType.add</code>	Specifies that an update adds a component.
<code>enumUpdateType.replace</code>	Specifies that an update replaces a component.
<code>enumUpdateType.remove</code>	Specifies that an update removes a component.
<code>enumUpdateType.patch</code>	Specifies that an update modifies a component, but does not change its version.

Enum Type: `enumEventResult`

Values:

0	Succeeded
1	Failed
2	Cancelled

Type Detail:

An event result instance denotes a particular result for an update event. The result indicates success, failure, or cancellation.

Value Detail:

<code>enumEventResult.succeeded</code>	Specifies that the operation was successful.
<code>enumEventResult.failed</code>	Specifies that the operation failed.
<code>enumEventResult.cancelled</code>	Specifies that the operation was cancelled.

Chapter 6. Notices

- Third party license terms and conditions, notices and information
- Accessing the product Web site
- Accessing the product documentation
- Trademarks

Third party license terms and conditions, notices and information

The relevant terms and conditions, notices and other information are provided in the "LICENSE.TXT" file on the root directory of the first installation CD-ROM for the product that the update installer updates. Please note that any non-English version of the information in this file is unofficial and is provided to you for your convenience only. The English version of the file is the official version.

Accessing the product Web site

To see the most updated product information, please go to the WebSphere Application Server Library page

Accessing product documentation

The following documentation is available when you use the update installer application:

readme_updateInstaller.txt, readme_updateInstaller.html and readme_updateInstaller.pdf files

You are currently viewing the `readme_updateInstaller` file. This file is available in three formats from the directory where you download and unpack the update installer application ZIP file or a fix pack ZIP file. They are identical files except for small formatting differences.

readme_was50_fpn.html

The `readme_was50_fpn.html` file is available on the Support Web site.

Click **All code fixes and support tools** under the "Software downloads" heading. Scroll to a fix pack download and click the link to display a page that describes the fix pack and has a link to the readme file.

The `readme_was50_fpn.html` file provides instructions for downloading fix pack *n*.

Release Notes

The *Release Notes* document describes changes to product documentation and workarounds for any problems that might exist in a fix pack. The *Release Notes* are available from the Support Web site.

List of fixes

The list of fixes describes every fix that is included in a fix pack. The list of fixes is available from the Support Web site.

Trademarks and service marks

The following terms are trademarks of IBM Corporation in the United States, other countries, or both:

- Everyplace
- iSeries
- IBM
- Redbooks
- ViaVoice
- WebSphere

- zSeries

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel Inside (logos), MMX and Pentium are trademarks of Intel 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.

SET and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product and service names may be trademarks or service marks of others.