IBM WebSphere Application Server

Readme for the update installer application for Version 5

Note

Before using this information, be sure to read the general information under Chapter 6, “Notices”, on page 75.

Compilation date: May 5, 2003

Contents

About this readme_updateinstaller document

This readme_updateinstaller document describes procedures for using the update installer application to install a 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 a fix or fix pack from these WebSphere Application Server products.

You can also use the update installer to install 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 a fix or fix pack is installed or removed.

All of the information in this document is available in the online Information Center at http:// publib7b.boulder.ibm.com/ webapp/ wasinfo1/ index.jsp? deployment= ApplicationServer.

Chapter 1. Installing fixes and fix packs

Install 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 commands to install 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.

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 fix packs 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 a fix or a fix pack in an IBM WebSphere Application Server, V5 environment, using the update installer application.

To apply a fix or fix pack, you must first set up and configure the environment by downloading the 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 you can apply the fix or fix pack, using either the wizard interface, updateWizard.sh or updateWizard.bat, or its silent, command-line interface, updateSilent.sh or updateSilent.bat.

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

If the base WebSphere Application Server node is within a cell, open this topic in the Network Deployment InfoCenter. If the base Application Server node is extended by the Enterprise product, open this topic in the Enterprise InfoCenter. This topic describes applying a fix or fix pack to a standalone base node.

Use the versionInfo or historyInfo commands in the `install_root/bin` directory, to display the exact fix and version level of the product. You can also use the silent update installer application to:

- View fix information
- View fix pack information

Note: 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 a 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 standalone base node by installing a fix or fix pack. The Uninstalling fixes and fix packs topic describes how to remove a fix or fix pack from a standalone base node.

Note: Installing a fix pack uninstalls all fixes. If some of the fixes that are uninstalled are a later level than the fix pack, their function is not included in the fix pack. You must 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. After unpacking the ZIP file you download, delete the ZIP file to free space. For a fix pack, have approximately 250 MB of free space in the `/tmp` directory on a UNIX-based platform, or on the disk drive where you are installing on a Windows platform.

Also, space is required for backup files. When installing a fix pack the space required is typically about the same as the size of the fix pack, that is, between 25 MB and 165 MB, depending on the particular fix pack.

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.
2. Apply the fix or fix pack to the base node.

   To apply the fix or fix pack, you must stop each server on the base node. You must also set up and configure the environment by downloading the fix or fix pack, downloading the update installer (if necessary), and creating update repositories. Use the following procedure to perform these tasks:
   a. Stop each base node with the **stopNode** command.
   b. Stop each server on the base WebSphere Application Server node with the **stopServer** command.

      Stop all WebSphere Application Server-related Java processes. On a Windows platform, you can use the task manager to stop Java processes. On a UNIX-based platform, use the **kill** command to stop Java processes.
   c. Set up and configure the environment.
      1) Download the update installer application zip file, if you are installing an interim fix.

         You should download the current version of the file, even though you might have an update installer from a previous fix installation. The Support page links to the current installer. All installers should be able to install older fixes and fix packs.
      2) Create an `install_root/update` directory, if it does not already exist.
      3) Extract the contents of the zip file to the `update` directory.
      4) 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.

5) Download the fix from the Support page to the fixes repository directory, if you are installing an interim fix. Download the fix pack JAR file to the install_root/update directory. Unpacking the fix pack automatically creates the fixpacks repository.

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

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

   The location of the update, fixes repository, and fixpacksrepository directories is arbitrary. You can 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 directory to the bin directory of the installation root.

   c) Run the appropriate command:

      - **.** *install_root***/bin/setupCmdLine.sh** (for an Application Server product)
      - *install_root***\bin\setupCmdLine.bat** (Windows platforms only)
      - **.** *install_root***/bin/setupClient.sh** (for the Application Server client)
      - *install_root***\bin\setupClient.bat** (Windows platforms only)

d. Apply the fix or fix pack to the base node.

   Depending on the interface you use to the update installer:

   - Refer to the updateWizard command topic for usage information.
   - Refer to the **updateSilent** command description for the proper syntax for installing the fix or fix pack:
     – Installing fixes
     – Installing fix packs

   For example, to install the was50_fp1_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_fp1_win
```

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

e. Restart each server on the node with the **startServer** command.

f. Restart the node agent for the base node with the **startNode** command.

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

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

   - Does the fix or fix pack appear in the wizard panel that lists applied fixes, or the panel that lists applied fix packs? If so, the fix or fix pack is installed.

- Does the fix or fix pack appear in the wizard panel that lists installable fixes, or the panel that lists installable fix packs? If so, the fix or fix pack is uninstalled.
- Is there a `[fixID].efix` or `[fix packID].ptf` in the `install_root/properties/version/version` directory, or a `[fixID].efixApplied`, `[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 fix information and update state show that the fix is installed or removed?
- Does collecting fix pack information and update state show that the fix pack is installed or removed?

You can also run the installation verification tool on the node as described in the online InfoCenter to ensure 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 fix packs 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 a fix or a fix pack (also known as a program temporary fix, or PTF) in an IBM WebSphere Application Server Network Deployment, V5 environment, using the update installer application.

To apply a fix or fix pack, you must first set up and configure the environment by downloading the 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 you can apply the fix or fix pack, using either the wizard interface, updateWizard.sh or updateWizard.bat, or its silent, command-line interface, updateSilent.sh or updateSilent.bat.

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

One requirement governs applying a fix or fix pack to a cell, to ensure 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 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 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.

You must ensure that the 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. You must uninstall a fix or fix pack from every base node, if you uninstall the fix or fix pack from the deployment manager node.

Use the versionInfo or historyInfo commands in the `install_root/bin` directory, to display the exact fix and version level of the product. You can also use the silent update installer application to:

- View fix information
- View fix pack information

Note: 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 a 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. You can 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, you must restart all node agents in the cell to ensure correct operation. This includes node agents on base nodes that you have not updated to V5.0.1.

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

Note: Installing a fix pack uninstalls all fixes. If some of the fixes that are uninstalled are a later level than the fix pack, their function is not included in the fix pack. You must 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. After unpacking the ZIP file you download, delete the ZIP file to free space. For a fix pack, have approximately 250 MB of free space in the `/tmp` directory on a UNIX-based platform, or on the disk drive where you are installing on a Windows platform.

Also, space is required for backup files. When installing a fix pack the space required is typically about the same as the size of the fix pack, that is, between 25 MB and 165 MB, depending on the particular fix pack.

Fixes require much less space to install.

Steps for this task

1. Stop the deployment manager process with the **stopManager** command.

   Stop all WebSphere Application Server-related Java processes.

2. Apply the fix or fix pack to the deployment manager node.

   To apply the fix or fix pack, you must first set up and configure the environment by downloading the fix and the update installer, or downloading the fix pack (which includes the update installer), and creating update repositories. Then you can use the update installer application to apply the fix or fix pack, using either its wizard interface or its silent, command-line interface. Use the following procedure to perform these tasks:

   a. Set up and configure the environment.

      The location of the `update`, `fixes` repository, and `fixpacks`repository directories in the following steps is arbitrary. You can create the directories anywhere. However, the *`install_root`*`/update`, *`install_root`*`/update/fixes`, and *`install_root`*`/update/fixpacks` locations are recommended.

      If you installed the WebSphere Application Server product you are updating in a non-standard location, you must set the JAVA_HOME environment variable in the optional configuration step that follows.

      1) Download the update installer application zip file, if you are installing an interim fix.

         You should download the current version of the file, even though you might have an update installer from a previous fix installation. The Support page links to the current installer. All installers should be able to install older fixes and fix packs.

      2) Create an `install_root/update` directory, if it does not already exist.

      3) Extract the contents of the zip file to the `update` directory.

      4) 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.

      5) Download the fix from the Support page to the `fixes` repository directory, if you are installing an interim fix. Download the fix pack JAR file to the `install_root/update` directory. Unpacking the fix pack automatically creates the `fixpacks` repository.

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

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

         The location of the `update`, `fixes` repository, and `fixpacks`repository directories is arbitrary. You can 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 directory to the `bin` directory of the installation root.

        c) Run the appropriate command:
            - **.** *install_root***/bin/setupCmdLine.sh** (for an Application Server product)
            - *install_root***\bin\setupCmdLine.bat** (Windows platforms only)

    b. Use the appropriate command to apply the fix or fix pack on the deployment manager node.

       Depending on the interface you use to the update installer:

       - Refer to the updateWizard command topic for usage information.
       - Refer to the **updateSilent** command description for the proper syntax for installing the fix or fix pack:
         – Installing fixes
         – Installing fix packs

       For example, to install the `was50_nd_fp1_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_fp1_win
```

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

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

4. Ensure that the deployment manager node is fully functional and has the fix or fix pack applied.

   Run the installation verification tool on the node to ensure that the node is operational.

   Use the versionInfo, historyInfo, or updateSilent command to view the fix or fix pack level.

5. Restart the node agent of each base node.

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

   There are several ways to restart a node agent:

   - Use the administrative console to restart (stop and start) all node agents. This is the most convenient method because you can restart one, more, or all node agents at one time.
   - Use the wsadmin command to stop a node agent.
   - Use the install_root/bin/stopNode command on each base node, to stop its node agent.
   - Use the install_root/bin/startNode command on each base node, to start its node agent.

6. Perform the following steps for each node to which you intend to apply the fix or fix pack.

   To apply the fix or fix pack, you must stop each server on the base node, including the nodeagent. Stop all WebSphere Application Server-related Java processes. You must also set up and configure the environment by downloading the fix or fix pack, downloading the update installer (if necessary), and creating update repositories. Use the following procedure to perform these tasks:

a. Stop each base node with the **stopNode** command.

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

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

c. Set up and configure the environment.

   1) Download the update installer application zip file, if you are installing an interim fix.

      You should download the current version of the file, even though you might have an update installer from a previous fix installation. The Support page links to the current installer. Any update installer can install older fixes and fix packs. The application is backward compatible.

   2) Create an `install_root/update` directory, if it does not already exist.

   3) Extract the contents of the ZIP file to the `update` directory.

   4) 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.

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

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

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

      The location of the `update`, `fixes` repository, and `fixpacks`repository directories is arbitrary. You can 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 directory to the `bin` directory of the installation root.

      c) Run the appropriate command:

         - **.** *install_root*/**bin/setupCmdLine.sh** (for an Application Server product)
         - *install_root***bin\setupCmdLine.bat** (Windows platforms only)
         - **.** *install_root*/**bin/setupClient.sh** (for the Application Server client)
         - *install_root***bin\setupClient.bat** (Windows platforms only)

d. Apply the fix or fix pack to the base product.

   Depending on the interface you use to the update installer:

   - Refer to the updateWizard command topic for usage information.

- Refer to the **updateSilent** command description for the proper syntax for installing the fix or fix pack:
  - Installing fixes
  - Installing fix packs

For example, to install the `was50_fp1_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_fp1_win
```

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

e. Restart each server on the node with the **startServer** command.

f. Restart the node agent for the base node with the **startNode** command.

g. Verify that the base node is fully functional and that it has the fix or fix pack applied.

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

   - Does the fix or fix pack appear in the wizard panel that lists applied fixes, or the panel that lists applied fix packs? If so, the fix or fix pack is installed.
   - Does the fix or fix pack appear in the wizard panel that lists installable fixes, or the panel that lists installable fix packs? If so, the fix or fix pack is uninstalled.
   - Is there a `[fixID].efix` or `[fix packID].ptf` in the `install_root/properties/version/version` directory, or a `[fixID].efixApplied`, `[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 fix information and update state show that the fix is installed or removed?
   - Does collecting fix pack information and update state show that the fix pack is installed or removed?

   You can also run the installation verification tool on the node to ensure that the node is operational.

7. Specify that file sets on each base node match those on the deployment manager node.

   Ensure consistent configuration data across a cell. You can 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**. You can 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.

8. Verify that all nodes are online and that the cell is functioning correctly.

9. 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 *fix packs* 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 a 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 a fix or fix pack, first set up and configure the environment by downloading the 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 you can use the update installer to install the 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 a fix or fix pack to a cell, to ensure 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 V5.0.1 base WebSphere Application Server node to a V5.0.0 deployment manager cell.
- **Requirement 2:** The Enterprise product must be at the same fix pack level as the product it extends:
  - If the Enterprise product extends a base WebSphere Application Server node, the 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 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 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 a fix or fix pack from every base node, if you uninstall the fix or fix pack from the deployment manager node.

Use the versionInfo or historyInfo commands in the `install_root/bin` directory, to display the exact fix and version level of the product. You can also use the silent update installer application to:

- View fix information
- View fix pack information

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

If you are applying the fix or fix pack to the Enterprise product on a stand-alone application server, follow step 13, which describes how to update a base node.

The Uninstalling fixes and fix packs topic describes how to remove a 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 fix or fix pack from each base node in a cell, and from the Enterprise product on each base node before you can uninstall the fix or fix pack from the deployment manager node, and from the Enterprise product on the deployment manager node.

Note: Installing a fix pack uninstalls all fixes. If some of the fixes that are uninstalled are a later level than the fix pack, their function is 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. After unpacking the ZIP file you download, delete the ZIP file to free space. For a fix pack, have approximately 250 MB of free space in the `/tmp` directory on a UNIX-based platform, or on the disk drive where you are installing on a Windows NT or Windows 2000 platform.

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 25 MB and 165 MB, depending on the particular fix pack.

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.

   If you do not have a deployment manager node, skip to step 13.
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` 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 fix installation. The Support page links to the current installer. All versions of the update installer application can install older V5 fixes and fix packs.
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.

   It is not necessary to create the `fixpacks` repository directory. Unpacking a fix pack creates the `fixpacks` directory if it does not already exist.
7. Download the fix or download and unpack the fix pack.

   Download a 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.

   Note: On Windows NT or Windows 2000 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.

   Note: If the update installer can set the Java environment, this step is unnecessary. Otherwise, this is a required step.

   The location of the `update`, `fixes` repository, and `fixpacks`repository directories is arbitrary. You can 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 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 command to set JAVA_HOME.

      Issue the appropriate command:

      - **.** *install_root***/bin/setupCmdLine.sh**(for an Application Server product)
      - *install_root***\bin\setupCmdLine.bat** (Windows NT or Windows 2000 platforms only)

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

   a. Use the appropriate command to apply the fix or fix pack on the deployment manager node.

      Depending on the interface you use to the update installer:

      - Refer to the updateWizard command topic for usage information.
      - Refer to the **updateSilent** command description of the proper syntax for installing the fix or fix pack:
        - Installing fixes
        - Installing fix packs

      For example, to install the was50_nd_fp1_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_fp1_win
```

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

   b. 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_fp1_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_fp1_win
```

      The command is shown here 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 each base node.

11. 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 a fix or fix pack:

    - Does the fix or fix pack appear in the wizard panel that lists applied fixes, or the panel that lists applied fix packs? If so, the fix or fix pack is installed.
    - Does the fix or fix pack appear in the wizard panel that lists installable fixes, or the panel that lists installable fix packs? If so, the fix or fix pack is uninstalled.
    - Is there a [fixID].efix or [fix packID].ptf in the install_root/properties/version/version directory, or a [fixID].efixApplied, [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 fix information and update state show that the fix is installed or removed?
- Does collecting fix pack information and update state show that the fix pack is installed or removed?

12. 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. You can restart all node agents, but you need not restart node agents on base nodes that you intend to update with the fix or fix pack. The 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 following steps for each base node to which you intend to apply the 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 NT or Windows 2000 platform, you can use the task manager to stop Java processes. On a UNIX-based platform, use the **kill** command to stop Java processes.

    c. Create an `install_root/update` directory, if it does not already exist.

    d. 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 fix installation. The Support page links to the current installer. All versions of the update installer application can install or remove older V5 fixes and fix packs.

    e. Extract the contents of the ZIP file to the `update` directory.

    f. Create the `update/fixes` repository if you are installing an interim fix.

       It is not necessary to create the `fixpacks` repository directory. Unpacking the fix pack creates the `fixpacks` directory, if it does not already exist.

    g. Download the fix or download and unpack the fix pack.

       Download a 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.

       Note: On Windows NT or Windows 2000 platforms, the pkunzip utility might not decompress the download image correctly. Use another utility (such as WinZip) to unzip the image.

    h. **(Optional)** Set up the Java environment for the update installer.

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

       1) Open a command-line window.

       2) Change directory to the `bin` directory of the installation root.

       3) Issue the appropriate command to set JAVA_HOME:

          - **.** *install_root*/**bin/setupCmdLine.sh** (for an Application Server product)

- *install_root***bin\setupCmdLine.bat** (Windows platforms only)
- **.** *install_root*/**bin/setupClient.sh** (for the Application Server client)
- *install_root***bin\setupClient.bat** (Windows platforms only)

i. Apply the fix or fix pack to the base node.

Depending on the interface to the update installer:

- Refer to the updateWizard command topic for usage information.
- Refer to the **updateSilent** command description of the proper syntax for installing the fix or fix pack:
  - Installing fixes
  - Installing fix packs

For example, to install the `was50_fp1_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_fp1_win
```

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

j. Install the fix or fix pack to the Enterprise product.

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.

For example, to install the `was50_pme_fp1_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_fp1_win
```

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

k. Restart each server on the node with the **startServer** command.

l. Restart the node agent for the base node with the **startNode** command.

m. Verify that the base node is fully functional and that it has the fix or fix pack applied.

There are several ways to verify the successful application of a fix or fix pack as described in the earlier step that verifies that the deployment manager node is functional.

14. Specify that file sets on each base node match those on the deployment manager node.

    Ensure consistent configuration data across a cell. You can 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**. You can 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.

The cell is now fully functional. All operations are available and functioning normally.

Chapter 2. Uninstalling fixes and fix packs

Install fixes and fix packs to 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

The update installer application installs and uninstalls fixes and fix packs (also known as fix packs and program temporary fixes, or PTFs) on WebSphere Application Server, Version 5 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 uninstalling a fix or a fix pack in an IBM WebSphere Application Server, V5 environment, using the update installer application.

To remove a fix or fix pack, you must first set up and configure the environment for the update installer, by running the **setupCmdLine** command if you use the silent interface with non-standard directories. You apply the fix or fix pack with either the wizard interface, updateWizard.sh or updateWizard.bat, or its silent, command-line interface, updateSilent.sh or updateSilent.bat.

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

If the base WebSphere Application Server node is within a cell, open this topic in the Network Deployment InfoCenter. If the base Application Server node is extended by the Enterprise product, open this topic in the Enterprise InfoCenter. This topic describes removing a fix or fix pack from a standalone base node.

Use the **versionInfo** command in the `bin` directory of the product installation root, to display the exact fix and version level of the product. You can also use the silent update installer application to:

- View fix information
- View fix pack information

Note: 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 a 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 standalone base node by removing a fix or fix pack. The ″Installing fixes and fix packs″ topic describes how to apply a fix or fix pack to a standalone base node.

Steps for this task

1. Remove the fix or fix pack from the base node.
   a. Stop each server on the base node with the **stopServer** command.
   b. **(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. You can 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** (for an Application Server product)
         - *install_root***\bin\setupCmdLine.bat** (Windows platforms only)
         - **.** *install_root***/bin/setupClient.sh** (for the Application Server client)
         - *install_root***\bin\setupClient.bat** (Windows platforms only)
   c. Use the appropriate command to remove the fix or fix pack from the base node.

      Depending on the interface you use to the update installer:
      - Refer to the updateWizard command topic for usage information.
      - Refer to the **updateSilent** command description for the proper syntax for uninstalling the fix or fix pack:
        – Uninstalling fixes
        – Uninstalling fix packs

      For example, to uninstall the `was50_fp1_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_fp1_win
```

      The command is shown on more than one line, for clarity.
   d. Restart each server on the node with the **startServer** command.
2. Verify that the node is online and functioning correctly.

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

- Does the fix or fix pack appear in the wizard panel that lists applied fixes, or the panel that lists applied fix packs? If so, the fix or fix pack is installed.
- Does the fix or fix pack appear in the wizard panel that lists installable fixes, or the panel that lists installable fix packs? If so, the fix or fix pack is uninstalled.
- Is there a `[fixID].efix` or `[fix packID].ptf` in the `install_root/properties/version/version` directory, or a `[fixID].efixApplied`, `[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 fix information and update state show that the fix is installed or removed?
- Does collecting fix pack information and update state show that the fix pack is installed or removed?

You can also run the installation verification tool on the node to ensure that the node is operational.

You can successfully remove fixes and fix packs from the WebSphere Application Server product.

Removing fixes and fix packs from the Network Deployment product

The update installer application installs and uninstalls fixes and fix packs (also known as fix packs and program temporary fixes, or PTFs) on WebSphere Application Server, V5 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 uninstalling a fix or a fix pack (also known as a program temporary fix, or PTF) in an IBM WebSphere Application Server Network Deployment, V5 environment, using the update installer application.

To remove a fix or fix pack, you must first set up and configure the environment for the update installer, by running the **setupCmdLine** command if you use the silent interface with non-standard directories. You apply the fix or fix pack with either the wizard interface, updateWizard.sh or updateWizard.bat, or its silent, command-line interface, updateSilent.sh or updateSilent.bat.

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

One requirement governs removing a fix or fix pack from a cell, to ensure 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 of a base WebSphere Application Server node at V5.0.1, to add it to a cell that is owned by a V5.0.0 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.

You must ensure that the 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. You must uninstall a 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 `bin` directory of the product install_root, to display the exact fix and version level of the product. You can also use the silent update installer application to:

- View fix information
- View fix pack information

Note: 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 a 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 a fix or fix pack from an entire Network Deployment cell, or from any part of the cell. According to the guidelines, you must 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.

The "Installing fixes and fix packs" topic describes how to apply a fix or fix pack to an entire cell, or to selected parts of the cell. According to the guidelines, you must apply a fix or fix pack to the deployment manager node, before you can apply the fix or fix pack to any base node in the cell.

Steps for this task

1. Remove the fix or fix pack from all base nodes.

   Determine the base nodes from which you intend to remove the fix or fix pack, and perform the following steps for each node.

   a. Stop each base node with the **stopNode** command.

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

   b. Stop each server on the base node with the **stopServer** command.

   c. Set up and configure the environment.

      The location of the `update`, `fixes` repository, and `fixpacks`repository directories in the following steps is arbitrary. You can create the directories anywhere. However, the *`install_root`*`/update`, *`install_root`*`/update/fixes`, and *install_root*/update/fixpacks locations are recommended. If you did not put the command into the update directory, you must run the optional configuration step.

1) If you do not have the update installer application, you can download it from the Support page.
2) Create an install_root/update directory, if it does not already exist.
3) Extract the contents of the update installer zip file to the update directory.

   Note: On Windows platforms, the pkunzip utility might not decompress the download image correctly. Use another utility (such as WinZip) to unzip the image.
4) **Optional:** Set the Java environment for the update installer.

   The location of the update, fixes repository, and fixpacksrepository directories is arbitrary. You can 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 directory to the bin directory of the installation root.
   c) Run the appropriate command:
      - **.** *install_root***/bin/setupCmdLine.sh** (for an Application Server product)
      - *install_root***\bin\setupCmdLine.bat** (Windows platforms only)
      - **.** *install_root***/bin/setupClient.sh** (for the Application Server client)
      - *install_root***\bin\setupClient.bat** (Windows platforms only)

d. Remove the fix or fix pack from the base node.

   Depending on the interface you use to the update installer:
   - Refer to the updateWizard command topic for usage information.
   - Refer to the **updateSilent** command description for the proper syntax for uninstalling the fix or fix pack:
     – Uninstalling fixes
     – Uninstalling fix packs

   For example, to uninstall the was50_fp1_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_fp1_win
```

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

e. Restart each server on the node with the **startServer** command.

f. Restart the node agent for the base node with the **startNode** command.

g. Verify that each base node is fully functional and has the fix or fix pack removed.

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

   - Does the fix or fix pack appear in the wizard panel that lists applied fixes, or the panel that lists applied fix packs? If so, the fix or fix pack is installed.
   - Does the fix or fix pack appear in the wizard panel that lists installable fixes, or the panel that lists installable fix packs? If so, the fix or fix pack is uninstalled.
   - Is there a `[fixID].efix` or `[fix packID].ptf` in the `install_root/properties/version/version` directory, or a `[fixID].efixApplied`, `[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 fix information and update state show that the fix is installed or removed?
   - Does collecting fix pack information and update state show that the fix pack is installed or removed?

   You can also run the installation verification tool on the node to ensure that the node is operational.

2. Stop the deployment manager with the **stopManager** command.
3. Remove the fix or fix pack from the Network Deployment node.

   Note: If you remove the fix or fix pack from every base node in the cell, you can also remove the 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. You can 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** (for an Application Server product)
         - *install_root***\bin\setupCmdLine.bat** (Windows platforms only)

   b. Remove the fix or fix pack from the deployment manager node.

Depending on the interface you use to the update installer:

- Refer to the updateWizard command topic for usage information.
- Refer to the **updateSilent** command description for the proper syntax for uninstalling the fix or fix pack:
  - Uninstalling fixes
  - Uninstalling fix packs

For example, to uninstall the `was50_nd_fp1_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_fp1_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 application of a fix or fix pack:

   - Does the fix or fix pack appear in the wizard panel that lists applied fixes, or the panel that lists applied fix packs? If so, the fix or fix pack is installed.
   - Does the fix or fix pack appear in the wizard panel that lists installable fixes, or the panel that lists installable fix packs? If so, the fix or fix pack is uninstalled.
   - Is there a `[fixID].efix`, `[fixID].efixApplied`, `[fixID].efixDriver`, `[fix packID].ptf`, `[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 fix information and update state show that the fix is installed or removed?
   - Does collecting fix pack information and update state show that the fix pack is installed or removed?

   You can also run the installation verification tool on the node to ensure that the node is operational.
6. Specify that file sets on one node match those on the deployment manager node.

   Ensure consistent configuration data across a cell. You can 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**. You can 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

The update installer application installs and uninstalls fixes and fix packs (also known as *fix packs* and *program temporary fixes*, or PTFs) on WebSphere Application Server, V5 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 a fix or a fix pack to an entire cell or to a single machine in an IBM WebSphere Application Server Enterprise, V5 environment, using the update installer application.

To remove a fix or fix pack, first set up and configure the environment for the update installer, by running the **setupCmdLine** command if you use the silent interface with non-standard directories. Then you can use the update installer to remove a 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 a fix or fix pack to a cell, to ensure 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 or fix pack level.

  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.
- **Requirement 2:** The Enterprise product must be at the same fix or fix pack level as the product it extends:
  - If the Enterprise product extends a base WebSphere Application Server node, the 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 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 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 a 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 a fix or fix pack from all base nodes:
   a. Remove the fix or fix pack from the Enterprise product that extends the base node.
   b. Remove the fix or fix pack from the base product on the node.

2. Remove the fix or fix pack from the Enterprise product that extends the deployment manager.
3. Remove the fix or fix pack from the deployment manager.

Use the **versionInfo** command in the `bin` directory of the product installation root, to display the exact fix and version level of the product. You can also use the silent update installer application to:

- View fix information
- View fix pack information

Note: 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 a 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 a 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.

If you are removing the fix or fix pack from the Enterprise product on a stand-alone application server, you can follow step 1, which describes how to update a base node.

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

Steps for this task

1. Remove the fix or fix pack from all base nodes.

   Determine the base nodes from which you intend to remove the fix or fix pack, and perform the following steps for each node:

   a. Stop each base node with the **stopNode** command.

      Stop all WebSphere Application Server-related Java processes. On a Windows NT or Windows 2000 platform, you can use the task manager to stop Java processes. On a UNIX-based platform, use the **kill** command to stop Java processes.

   b. Stop each server on the base node with the **stopServer** command.

   c. Set up and configure the environment.

      The location of the `update`, `fixes` repository, and `fixpacks`repository directories in the following steps is arbitrary. You can create the directories anywhere. However, the *install_root*/`update`, *install_root*/`update/fixes`, and *install_root*/`update/fixpacks` locations are recommended. If you did not put the command into the `update` directory, run the optional configuration step.

1) If you do not have the update installer application, you can download it from the Support page.

2) Create an `install_root/update` directory, if it does not already exist.

3) Extract the contents of the update installer zip file to the `update` directory.

   Note: On Windows NT or Windows 2000 platforms, the pkunzip utility might not decompress the download image correctly. Use another utility (such as WinZip) to unzip the image.

4) **Optional:** Set the Java environment for the update installer.

   The location of the `update`, `fixes` repository, and `fixpacks`repository directories is arbitrary. You can 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 directory to the `bin` directory of the installation root.

   c) Run the appropriate command:

      - **.** *install_root***/bin/setupCmdLine.sh** (for an Application Server product)
      - *install_root***\bin\setupCmdLine.bat** (Windows platforms only)
      - **.** *install_root***/bin/setupClient.sh** (for the Application Server client)
      - *install_root***\bin\setupClient.bat** (Windows platforms only)

d. Remove the fix or fix pack from the Enterprise product.

   For example, to remove the `was50_pme_fp1_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_fp1_win
```

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

e. Remove the fix or fix pack from the base node.

   Depending on the interface you use to the update installer:

   - Refer to the updateWizard command topic for usage information.
   - Refer to the **updateSilent** command description of the proper syntax for uninstalling the fix or fix pack:
     - Uninstalling fixes
     - Uninstalling fix packs

   For example, to uninstall the `was50_fp1_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_fp1_win
```

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

f. Restart each server on the node with the **startServer** command.

g. Restart the node agent for the base node with the **startNode** command.

h. Verify that each base node is fully functional and has the fix or fix pack removed.

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

   - Does the fix or fix pack appear in the wizard panel that lists applied fixes, or the panel that lists applied fix packs? If so, the fix or fix pack is installed.
   - Does the fix or fix pack appear in the wizard panel that lists installable fixes, or the panel that lists installable fix packs? If so, the fix or fix pack is uninstalled.
   - Is there a `[fixID].efix` or `[fix packID].ptf` in the `install_root/properties/version/version` directory, or a `[fixID].efixApplied`, `[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 fix information and update state show that the fix is installed or removed?
   - Does collecting fix pack information and update state show that the fix pack is installed or removed?

   You can also run the installation verification tool on the node to ensure that the node is operational.

2. Stop the deployment manager with the **stopManager** command.

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

3. Remove the fix or fix pack from the Network Deployment node.

   Note: If you remove the fix or fix pack from every base node in the cell, you can also remove the 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. You can 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** (for an Application Server product)
   - *install_root***\bin\setupCmdLine.bat** (Windows platforms only)

b. Use the appropriate command to remove the fix or fix pack from the Enterprise node.

Depending on the interface you use to the update installer:

- Refer to the updateWizard command topic for usage information.
- Refer to the **updateSilent** command description of the proper syntax for uninstalling the fix or fix pack:
  - Uninstalling fixes
  - Uninstalling fix packs

For example, to remove the `was50_pme_nd_fp1_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_fp1_win
```

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

c. Remove the fix or fix pack from the deployment manager node.

For example, to uninstall the `was50_nd_fp1_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_fp1_win
```

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

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 application of a fix or fix pack:

   - Does the fix or fix pack appear in the wizard panel that lists applied fixes, or the panel that lists applied fix packs? If so, the fix or fix pack is installed.
   - Does the fix or fix pack appear in the wizard panel that lists installable fixes, or the panel that lists installable fix packs? If so, the fix or fix pack is uninstalled.
   - Is there a `[fixID].efix`, `[fixID].efixApplied`, `[fixID].efixDriver`, `[fix packID].ptf`, `[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 fix information and update state show that the fix is installed or removed?
- Does collecting fix pack information and update state show that the fix pack is installed or removed?

   You can also run the installation verification tool on the node to ensure that the node is operational.

6. Specify that file sets on one node match those on the deployment manager node.

   Ensure consistent configuration data across a cell. You can 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**. You can 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.

   The cell is now fully functional. All operations are available and functioning normally.

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.

Note: 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 a 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.

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 NT and Windows 2000: `*drive*\Program Files\WebSphere\AppServer`
- Network Deployment product:
  - AIX platforms: `/usr/WebSphere/DeploymentManager`
  - Other UNIX and Linux platforms: `/opt/WebSphere/DeploymentManager`
  - Windows NT and Windows 2000: `*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 NT and Windows 2000: `*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 NT and Windows 2000: `*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 250 MB of free space in the `/tmp` directory on a UNIX-based platform, or on the disk drive where you are installing on a Windows platform.

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 25 MB and 165 MB, depending on the particular fix pack.

Fixes require much less space to install.

The update installer checks for required space before it installs a fix or fix pack.

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. Set the environment variable or issue the appropriate command script, from the **/bin** directory of the installation root:

- . *install_root*/**bin/setupCmdLine.sh** (for an Application Server product)
- *install_root***bin\setupCmdLine.bat** (Windows platforms only)
- . *install_root*/**bin/setupClient.sh** (for the Application Server client)
- *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

Product	Operating system platform	Fix Pack 1 ZIP file	Fix Pack 1 ID	Default repository in installation root directory
Base WebSphere Application Server	AIX	was50_ fp1_aix.zip	was50_ fp1_aix	`../update/ fixpacks`
	Linux	was50_ fp1_linux.zip	was50_ fp1_linux	
	Linux/390	was50_ fp1_linux390.zip	was50_ fp1_linux390	
	Solaris	was50_ fp1_solaris.zip	was50_ fp1_solaris	
	Windows NT and Windows 2000	was50_ fp1_win.zip	was50_ fp1_win	`..\update\ fixpacks`

Table 1. Fix pack names for Fix Pack 1 (continued)

Product	Operating system platform	Fix Pack 1 ZIP file	Fix Pack 1 ID	Default repository in installation root directory
Network Deployment	AIX	was50_nd_ fp1_aix.zip	was50_nd_ fp1_aix	`../update/ fixpacks` 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.
	Linux	was50_nd_ fp1_linux.zip	was50_nd_ fp1_linux	
	Linux/390	was50_nd_ fp1_linux390.zip	was50_nd_ fp1_linux390	
	Solaris	was50_nd_ fp1_solaris.zip	was50_nd_ fp1_solaris	
	Windows NT and Windows 2000	was50_nd_ fp1_win.zip	was50_nd_ fp1_win	`..\update\ fixpacks`
Enterprise	AIX	was50_pme_ fp1_aix.zip	was50_pme_ fp1_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_ fp1_aix (to extend the Network Deployment product)	
	Linux	was50_pme_ fp1_linux.zip	was50_pme_ fp1_linux (base)	
			was50_pme_nd_ fp1_linux (Network Deployment)	
	Linux/390	was50_pme_ fp1_linux390.zip	was50_pme_ fp1_linux390 (base)	
			was50_pme_nd_ fp1_linux390 (Network Deployment)	
	Solaris	was50_pme_ fp1_solaris.zip	was50_pme_ fp1_solaris (base)	
			was50_pme_nd_ fp1_solaris (Network Deployment)	
	Windows NT and Windows 2000	was50_pme_ fp1_win.zip	was50_pme_ fp1_win (base)	`..\update\ fixpacks`
			was50_pme_nd_ fp1_win (Network Deployment)	

Table 1. Fix pack names for Fix Pack 1 (continued)

Product	Operating system platform	Fix Pack 1 ZIP file	Fix Pack 1 ID	Default repository in installation root directory
Express	AIX	was50_express_ fp1_aix.zip	was50_express_ fp1_aix	`../update/ fixpacks` 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_ fp1_linux.zip	was50_express_ fp1_linux	
	Linux/390	was50_express_ fp1_linux390.zip	was50_express_ fp1_linux390	
	Solaris	was50_express_ fp1_solaris.zip	was50_express_ fp1_solaris	
	Windows NT and Windows 2000	was50_express_ fp1_win.zip	was50_express_ fp1_win	`..\update\fixpacks`
WebSphere Application Server client	AIX	was50_client_ fp1_aix.zip	was50_client_ fp1_aix	`../update/ fixpacks` 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_ fp1_linux.zip	was50_client_ fp1_linux	
	Linux/390	was50_client_ fp1_linux390.zip	was50_client_ fp1_linux390	
	Solaris	was50_client_ fp1_solaris.zip	was50_client_ fp1_solaris	
	Windows NT and Windows 2000	was50_client_ fp1_win.zip	was50_client_ fp1_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, you can accept default fix and fix pack file locations, when using the **updateWizard** interface. Otherwise, you must browse to locate the fixes or fix packs you are installing or uninstalling.

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. A newer version can manage previously downloaded fixes and fix packs. 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_fp1_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 the`install_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.

Create an `update` directory if it does not exist, download the update installer application ZIP file from the Support site at http://www-3.ibm.com/software/webservers/appserv/support.html to the `update` directory, and unzip the file.

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

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 directory to the `bin` directory of the install_root.
3. Run the appropriate command:
   - **.** *install_root***/bin/setupCmdLine.sh** (for an Application Server product)
   - *install_root***\bin\setupCmdLine.bat** (Windows platforms only)
   - **.** *install_root***/bin/setupClient.sh** (for the Application Server client)
   - *install_root***\bin\setupClient.bat** (Windows platforms only)

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

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

The following examples describe various usage syntaxes. In each syntax example, optional parameters are enclosed by brackets ([]). Values that you must 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.

Note: 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 NT and Windows 2000 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.

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

-fixDetails
Fix only: Displays fix detail information.

-fixDir
Fix only: Specifies the fully qualified directory where you download fixes. This directory is usually the *install_root*/update/fixes directory.

-fixes Fix only: Specifies a list of space-delimited fixes to install or uninstall.

-fixJars
Fix only: Specifies a list of space-delimited fix JAR files to install or uninstall. Each JAR file has one or more 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_fp1_linux
- **fix pack JAR file name:** was50_fp1_linux.jar
- **fix pack ZIP file name:** was50_fp1_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: Specifies to skip 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
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.

You can use the templated `.properties` file 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_fp1_win
   fixpackDetails=true
   ihsInstallDir=C:\\IBMHttpServer
```

-skipIHS
fix pack only: Specifies no application of any optional service for the IBM HTTP Server product that might exist in the fix pack.

Note: If you installed the IBM HTTP Server product as a feature, use the update installer to update it with service in a fix or fix pack. Otherwise, you must 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.

You must 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.

-skipMQ
fix pack only: Specifies no application of any optional service for the embedded messaging feature (which is based on the IBM WebSphere MQ product) that might exist in the fix pack.

Note: 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 a fix or fix pack. You can skip the installation of service to the embedded messaging feature if you must install corrective service to the stand-alone IBM WebSphere MQ product.

-uninstall
Specifies to uninstall the identified fix or fix pack.

-uninstallAll
Specifies to uninstall all applied fixes. This parameter does not uninstall fix packs.

-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 fixes
- Uninstalling fixes
- Viewing information about fixes
- Installing fix packs
- Uninstalling fix packs
- Viewing information about fix packs

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

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 fixes

To install a collection of 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 fixes, and display 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 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 fixes from a Java archive (JAR) file, and display 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 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 fixes

To uninstall a collection of 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 fixes, and display 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 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 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 fixes in a Java archive (JAR) file, and display 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 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 fixes

To view a list of installed fixes:

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

To view a list of 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_fp1_win
```

To install a fix pack, and display fix 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_fp1_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_fp1_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_fp1_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_fp1_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_fp1_win
```

To uninstall a fix pack, and display fix details:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fixpack
         -installDir "C:\Program Files\WebSphere\AppServer"
         -uninstall
         -fixpackID was50_fp1_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 fixes and fix packs for WebSphere Application Server products. This topic describes the wizard interface to the update installer command, and gives some information about its panels and fields.

Note: 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 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 a 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.

See the updateSilent command topic for a description of update installer characteristics.

Create an `update` directory, download the update installer application ZIP file from the Support Web site at http://www-3.ibm.com/software/webservers/appserv/support.html to the `update` directory, and unzip the file.

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

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.

```
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.

-dpInstall
Disables installation prerequisite error locking.

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

The default behavior of the update installer application is to prevent further action if prerequisites for fixes are not met.

-dpUninstall
Disables uninstallation prerequisite error locking.

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

The default behavior of the update installer application is to prevent further action if prerequisites for fixes are not met.

-fixOnly
Allows fix installation and uninstallation only.

The default action for the update installer is to enable both 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 are installing 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. To install and uninstall fixes, the **updateWizard** does not require the local copy of the IBM SDK. This option bypasses making a local copy of the IBM SDK.

-usage Displays listing of optional parameters and usage to help you remember command syntax.

Displaying usage information

This command displays information about command syntax.

```
updateWizard -usage
```

Bypassing the local copy of the IBM SDK

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

```
updateWizard -fixOnly
```

Disabling prerequisite locking

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

To disable prerequisite locking when installing fixes:

```
updateWizard -fixOnly -dpInstall
```

To disable prerequisite locking when uninstalling fixes:

```
updateWizard -fixOnly -dpInstall
```

To disable prerequisite locking when installing and when uninstalling 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
- **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 at http://www-3.ibm.com/software/webservers/appserv/support.html. (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

directory input field. After selecting a product, its directory location appears in the **input field** for verification purposes. To make corrections or enter another product location, click **Specify product location**.

Menu panel

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

Fix repository specifier panel

Use this panel to provide the 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 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.

Note: About installation status

A 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 fix or fix pack updates to product components.

Installed status implies that the 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 fix or fix pack has at least one more update you can 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 a 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.
- You select to skip 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.
- You trigger a dynamic change in fix or fix pack status because:
  1. You apply all changes in a 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 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 fixes for uninstalling. Available fixes do not appear in the list. Only installed 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.

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 and pre-uninstallation summary panels

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

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

Installation and uninstallation

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 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 go back and install additional 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 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.

Note: See the note in the Installable fix selection panel section 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 a feature has required service, the feature appears in the list but is grayed out. If you did not install a feature during the original product installation, the feature does not appear. If there are no serviceable features in the fix pack, this panel does not appear. 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 a fix or fix pack. Otherwise, you must 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.

You must 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 a fix or fix pack. Do not select the installation of service to the embedded messaging feature if you must install corrective service to the stand-alone IBM WebSphere MQ product.

Pre-installation 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

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.

Post installation 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 [fixID].efixApplied, [fixID].efixDriver, [fix packID].ptfApplied, and [fix packID].ptfDriver files. A driver file has useful information about the entire contents of a 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.

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

XML files in the /properties/version directory

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 a 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 NT or Windows 2000 platforms.

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 fixes to the report.

-efixDetail
Adds details about applied 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.

genVersionReport.bat
Generates the `versionReport.html` report file in the `bin` directory on Windows NT or Windows 2000 platforms. The report includes the list of components, 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, 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. You can also 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.

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

***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

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`

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`

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`

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`

Backup JAR file names

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

For example: `20020924_211832_apar6789c_ras_undo.jar`

Note: 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`

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 a fix file (with an extension of `.efix`) to the `version` directory to indicate that a fix is currently installed.
- A WebSphere Application Server product removes a fix file from the `version` directory when it uninstalls the corresponding interim fix.
- A WebSphere Application Server product adds a fix driver file (with an extension of `.efixDriver`) to the `history` directory when a fix is installed. A fix driver file contains defining information for an interim fix.
- A WebSphere Application Server product removes a fix driver file when it removes the corresponding interim fix.
- A WebSphere Application Server product adds a fix application file (with an extension of `.efixApplied`) to the `history` directory when it installs an interim fix. A 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 a 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 than 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 a 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 a 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 a 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 a 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-proprty      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, you can omit certain potential component updates, but
only when the corresponding component is not installed.

You must 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-proprty       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 (a 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 a fix, usually indicating that the fix provides a 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:

A fix prerequisite instance denotes a 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,
                           you must install the fix before installing
                           the parent interim fix.  If true, you must not install
                           the fix before you install the parent interim fix.

fix-prereq.install-index   An optional index number which is 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 a 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 a 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
a 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 a 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:

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.

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:

enumEventResult.succeeded  Specifies that the operation was successful.

enumEventResult.failed     Specifies that the operation failed.

enumEventResult.cancelled  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:

http://www-3.ibm.com/software/webservers/appserv/was/library/

Accessing the 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_fp*n*.html
: The `readme_was50_fp`*`n`*`.html` file is available on the Support Web site at:

  http://www-3.ibm.com/software/webservers/appserv/support.html . 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_fp`*`n`*`.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.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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

Printed in U.S.A.