



## Readme

**Note**

Before using this information, be sure to read the general information under Appendix A, "Notices," on page 59.

**Compilation date: April 28, 2006**

**© Copyright International Business Machines Corporation 2004, 2006. All rights reserved.**

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

---

# Contents

**How to send your comments . . . . . v**

**Chapter 1. Installing maintenance packages . . . . . 1**

install.txt . . . . . 6  
  Location of the response file . . . . . 6  
  Installing silently . . . . . 7  
  Response file user entry validation . . . . . 7  
  Usage notes . . . . . 8  
  Example install.txt file . . . . . 8

**Chapter 2. Uninstalling maintenance packages. . . . . 11**

uninstall.txt . . . . . 14  
  Location of the response file. . . . . 14  
  Uninstalling silently . . . . . 14  
  Response file user entry validation . . . . . 15  
  Usage notes . . . . . 16  
  Example uninstall.txt file . . . . . 16

**Chapter 3. update command . . . . . 19**

Command options . . . . . 19  
  Commands for installing interim fixes . . . . . 19  
  Commands for uninstalling interim fixes . . . . . 20  
Logging . . . . . 21  
  Logs created when installing service . . . . . 21  
  Logs created when uninstalling service . . . . . 21  
  Indicators of success . . . . . 22  
Known problems and workarounds for the update command . . . . . 22

**Chapter 4. versionInfo command . . . . . 23**

Location of the command file . . . . . 23  
Syntax for the versionInfo command . . . . . 23  
Parameters . . . . . 24  
Report description . . . . . 24  
  Installation information . . . . . 24  
  Product list information . . . . . 25  
  Installed product information . . . . . 25  
Sample versionInfo report . . . . . 26  
Summary of Version 6 changes for the versionInfo command . . . . . 28  
  Changes to command syntax . . . . . 28  
  Changes to report information . . . . . 28

**Chapter 5. historyInfo command. . . . . 31**

Location of the command file . . . . . 31  
Syntax for the historyInfo command . . . . . 31  
Parameters . . . . . 32

Report description . . . . . 32  
  Installation information . . . . . 32  
  Installation event information . . . . . 33  
  Component installation event information . . . . . 33  
The event.history file . . . . . 33  
Summary of Version 6 changes for the historyInfo command . . . . . 35  
  Changes to command syntax . . . . . 35  
  Changes to report information . . . . . 35  
  Changes to the event.history file . . . . . 36

**Chapter 6. genHistoryReport command 37**

Location of the command file . . . . . 37  
Syntax for the genHistoryReport command. . . . . 37  
Report description . . . . . 38  
  Installation information . . . . . 38  
  Installation event information . . . . . 38  
  Component installation event information . . . . . 38  
The event.history file . . . . . 39  
Summary of Version 6 changes for the historyInfo command . . . . . 41  
  Changes to command syntax . . . . . 41  
  Changes to report information . . . . . 41  
  Changes to the event.history file . . . . . 41

**Chapter 7. genVersionReport command 43**

Location of the command file . . . . . 43  
Syntax for the genVersionReport command. . . . . 43  
Report description . . . . . 43  
  Installation information . . . . . 44  
  Product list information . . . . . 44  
  Installed product information . . . . . 44  
Summary of Version 6 changes for the VersionInfo command . . . . . 45  
  Changes to command syntax . . . . . 45  
  Changes to report information for Version 6.x . . . . . 45

**Chapter 8. Product version information 47**

Product information files . . . . . 47  
Reports. . . . . 48  
Logs and component backups . . . . . 49  
Directory locations . . . . . 50  
Operational description . . . . . 50  
Data dictionary . . . . . 51

**Appendix A. Notices . . . . . 59**

**Appendix B. Trademarks and service marks . . . . . 61**



---

## How to send your comments

Your feedback is important in helping to provide the most accurate and highest quality information.

- To send comments on articles in the WebSphere Application Server information center:
  1. Display the article in your Web browser and scroll to the end of the article.
  2. Click on the **Feedback** link at the bottom of the information center article, and a separate window containing an e-mail form appears.
  3. Fill out the e-mail form as instructed, and click on **Submit feedback** .
- To send comments on PDF books or readme files, you can e-mail your comments to: **wasdoc@us.ibm.com** or fax them to 919-254-0206.

Be sure to include the document name and number, the WebSphere Application Server version you are using, and, if applicable, the specific page, table, or figure number on which you are commenting.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.



---

## Chapter 1. Installing maintenance packages

This topic describes how to use the IBM Update Installer for WebSphere Software to install interim fixes, fix packs, and refresh packs. The Update Installer for WebSphere Software is also known as the Update Installer program, the UpdateInstaller program, and the Update installation wizard.

Use the proper authorizations to successfully install product updates.

When administrative security is enabled on WebSphere Application Server Network Deployment, for example, you must supply the administrative user ID and password before you can update the files.

Use the Update Installer program from the same installer ID that installed the product that you are updating. Otherwise, the file ownership mismatches might require correction by the root user.

### Important:

- The user account that originally installed the WebSphere Application Server product to be updated should be used to launch the Installation Wizard to install the Update Installer, and the same user account should be used to launch the Update Installer program to update a product.
  - When a different user account uses the *updi\_root* location, that user account must have reading and running access to that location. It must also have writing access to the *updi\_root/logs* directory and its subdirectory.
  - When a different user account is used to update the target WebSphere Application Server product location, that user account must have full access (reading, writing, and running) to the target location where a maintenance package is to be applied.
-  If a non-root user launches the Update Installer program, that user account must be able to run the **slibclean** command; otherwise, a root user must run the **slibclean** command whenever the Update Installer program is used. For more information, refer to the following Web page: <http://www-128.ibm.com/developerworks/eserver/tutorials/linking102/linking102-4-1.html>.
- Make sure that no processes from any users are locking any files in the target location where a maintenance package is to be installed.

The Update Installer wizard is an InstallShield for Multiplatforms wizard that runs with either a graphical user interface or in silent mode with a response file.

The following descriptions contain reference information about installing interim fixes, fix packs, and refresh packs on WebSphere Application Server products and components:

### Overview of the installation procedure

1. Download, unpack, and install the Update Installer for WebSphere Software; or install the Update Installer that is on the WebSphere Application Server supplements disc.
2. Download the most current version of the interim fix, fix pack, or refresh pack file from the Support site.

3. Download the interim fix, fix pack, or refresh pack from the Support Web site into the maintenance directory.
4. Use the Update Installer to install the interim fix, fix pack, or refresh pack.

The Update Installer creates a backup file in the *app\_server\_root*properties/version/nif/backup directory.

IBM does not support restoring a backup file that you have modified.

### Updating existing profiles in WebSphere Application Server products

The Update Installer updates the core product files in a WebSphere Application Server product. Service in a maintenance package might update the following files in the installation root directory:

- The SDK, Java technology edition, in the java/jre directory
- JAR files in the lib directory
- Scripts in the bin directory
- Profile templates

Some maintenance packages provide required service for existing profiles in addition to service for the core product files. Each maintenance package that has profile maintenance provides a script that changes the profile. The Update Installer prompts you to back up your configuration when installing a maintenance package that has required maintenance for profiles.

Some maintenance packages provide optional service for existing profiles. The readme file for the maintenance package describes whether the maintenance package contains optional service for existing profiles. If so, the readme file describes how to use the script provided with the maintenance package.

Use the **backupConfig** command to back up the configuration of each profile that the maintenance package can update. Or archive the *app\_server\_root*/profiles directory to back up all of the profiles at once.

If you uninstall a maintenance package, the Update Installer does not uninstall the maintenance package from profiles. The reason for not removing the maintenance is that you might have configured the profile after installing the maintenance. To restore an original profile, use the **restoreConfig** command or copy the profile from the archived *profile\_root* directory to replace the changed profile.

### Viewing the fix level of the product

Use the Chapter 4, “versionInfo command,” on page 23 and the Chapter 5, “historyInfo command,” on page 31 in the bin directory of the installation root directory to display the exact fix and version level of the product. However, do not use either command while installing or uninstalling a maintenance package.

### Updating cluster members

Apply the same maintenance packages to all of the WebSphere Application Server installations in a cluster. When all of the cluster members are not at the same service level, the following exception can occur:

```
DRSCacheApp E DRSW0008E:  
Exception is: com.ibm.disthub.impl.jms.JMSWrappedException:  
{-1361012295|unknown|java.io.OptionalDataException|}
```

This error can cause memory replication to not function properly.

**Required information.** The graphical interface requires you to supply the following information:

Table 1. Information required when installing a maintenance package

Field	Valid values	Description
File path of the installation root directory of the WebSphere product or component.  Installation root directory of the Update Installer.  See <i>updi_root</i> for more information.	Identify the installation root directory for one of the following products: <ul style="list-style-type: none"> <li>• IBM WebSphere Application Server</li> <li>• IBM WebSphere Application Server - Express</li> <li>• Embedded version of the IBM WebSphere Application Server - Express</li> <li>• IBM WebSphere Application Server Network Deployment</li> <li>• IBM WebSphere Extended Deployment</li> <li>• IBM Application Client for WebSphere Application Server</li> <li>• IBM WebSphere Business Integration Server Foundation</li> <li>• Web server plug-ins for WebSphere Application Server</li> </ul>	Download, unpack, and install the Update Installer for WebSphere Software. Or install the Update Installer that is on the WebSphere Application Server supplements disc. Install the Update Installer into each component that you intend to update.  The Update Installer application updates the product in its parent directory by default.
File name of the maintenance package to install.	Select a maintenance package to install from the maintenance directory.	The default maintenance package is the package with the latest date stamp and time stamp.

**Important:** See “Known problems and workarounds for the update command” on page 22 and the release notes for your product for the latest information about the Update Installer.

The following procedure describes how to install a maintenance package. See Chapter 2, “Uninstalling maintenance packages,” on page 11 for a description of how to roll back a maintenance package.

1. Log on as root on a Linux or UNIX operating system, or log on as a member of the administrator group on a Windows system.

 In addition, verify that the umask setting is 022. To verify the umask setting, issue the following command:

```
umask
```

To set the umask setting to 022, issue the following command:

```
umask 022
```

2. Install the product that you intend to update.  
You have very likely already installed the software that you are now updating. But if not, install the software now.
3. **Optional:** Install a new version of the Update Installer.  
Back up and uninstall any older copy of the Update Installer before downloading and installing the current Update Installer. To use a newer version of the Update Installer, you must first remove the older version.
  - a. Back up any files and subdirectories in the *updi\_root/maintenance* directory if necessary.
  - b. Uninstall the older version of the Update Installer using the program under *updi\_root/uninstall*.

- c. Download, unpack, and install the Update Installer for WebSphere Software; or install the Update Installer that is on the WebSphere Application Server supplements disc.
4. Download the maintenance package \*.pak file from the Support Web site into the maintenance directory.

Download maintenance packages for the Update Installer for WebSphere Software to install from the following IBM Web pages:

- Download maintenance packages for IBM WebSphere Extended Deployment from the IBM Support site for IBM WebSphere Extended Deployment.
- Download maintenance packages for Version 6.x of the WebSphere Application Server products from the IBM Support site for WebSphere Application Server.

**Tip:** Do not attempt to unzip or unpack the \*.pak file.

5.  Use the Windows Services panel to stop all services for WebSphere Application Server processes.
6. Stop all Java processes that use the IBM Software Developer Kit (SDK) that the WebSphere Application Server product provides.

Before installing or uninstalling interim fixes, fix packs, and refresh packs on a machine, stop all Java processes on the machine that use the IBM SDK, Java Technology Edition that WebSphere Application Server provides.

Stop all WebSphere Application Server-related Java processes that are running on the system where you are using the Update Installer program. For example, Java processes can include:

- All Java Virtual Machines (JVMs)
- WebSphere Application Server processes:  
WebSphere Application Server processes include:
  - Application server processes
  - The nodeagent process on an application server node when the node is federated into a deployment manager cell
  - The dmgr process for the deployment manager server
- IBM HTTP Server processes
- Web server processes that use a plug-in that you are updating
- First steps consoles
- Installation verification test (IVT) processes
- The Profile Management tool
- Other InstallShield for Multiplatforms (ISMP) installation programs
- InstallShield for Multiplatforms uninstall programs
- The IBM Rational Application Developer Agent Controller

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

See the following technote for more information, Stop all WebSphere Application Server-related Java processes before using the Update Installer for WebSphere software.

7. Verify that the following prerequisite conditions are met:
  - All of the product hardware and software prerequisites exist.  
The official statement of supported hardware and software is on the Supported hardware and software Web site.

- The WebSphere software that you are updating is correctly installed and is not corrupt.
  - The WebSphere SDK, Java technology edition is not corrupt.
  - The user is root on a Linux or UNIX system or a member of the administrator group on a Windows system.
8. Change directories to the *updi\_root* directory and use the update command to install the maintenance package.

Install the maintenance package on the deployment manager node before installing the maintenance package on each application server node that you intend to update.

Use the following command syntax to install the last maintenance package that you downloaded. The Update Installer wizard runs as a background process and does not display the graphical user interface when running in silent mode:

**Windows**

```
update.bat -silent -options responsefile
```

**Linux**

```
./update.sh -silent -options responsefile
```

Or, issue the update command to start the graphical user interface: **Windows**

```
update.bat
```

**Linux**

```
./update.sh
```

The following tables show options that are available when using the **update** command.

The commands in the first table each start the Update Installer wizard with a graphical user interface. The command in the second table causes the Update Installer wizard to run in silent mode.

Table 2. Update installer commands for installing with the graphical interface

Command example	Type of installation	Description
update.bat	Graphical interface mode	Initializes the maintenance package field with the name of the maintenance package that has the most recent date stamp and time stamp.  Accept all of the default values to install the maintenance package with the most recent time stamp.
update.bat -options "responsefiles/ <i>file_name</i> "	Graphical interface mode with an options file	Overrides all graphical interface values with values that you specified in the options response file.  Always use a response file that is based on the response file under <i>updi_root</i> /responsefiles.
update.bat -W maintenance.package="e:\IBM\WebSphere\AppServer\updateinstaller\maintenance\PQ20029.pak"	Graphical interface mode	Overrides the name of the maintenance package to apply.
update.bat -W product.location="e:\IBM\WebSphere\AppServer"	Graphical interface mode	Overrides the location of the WebSphere software to update.

Table 2. Update installer commands for installing with the graphical interface (continued)

Command example	Type of installation	Description
update.bat -W product.location="e:\IBM\WebSphere\AppServer" -W maintenance.package="e:\IBM\WebSphere\AppServer\updateinstaller\maintenance\PQ20029.pak"	Graphical interface mode	Overrides the location of the WebSphere software to update and the name of the maintenance package to apply.

Specify an appropriate JOBQ parameter value to have the job run in a different subsystem. Verify that the storage pool that the job runs in has as much memory as possible.

The command in the following table starts the Update Installer wizard in silent mode without the graphical user interface:

Table 3. Update Installer command for installing in silent mode

Command example	Type of installation	Description
update.bat -silent -options "responsefiles/file_name"	Silent mode with an options file	Overrides all default values with values that you specified in the options response file.  Always use a response file that is based on the response file under <i>updi_root</i> /responsefiles.

This procedure results in installing maintenance packages to update WebSphere software.

Click **Relaunch** on the last panel of the Update Installer to begin installing a second maintenance package.

After installing all maintenance packages, continue to use your WebSphere software.

---

## install.txt

The Update Installer for WebSphere Software can use an options response file to install maintenance packages from a command line interface.

The install.txt file has one directive that identifies the backup file for installing a service update. Comments in the file describe how to set the string value.

The Update Installer for WebSphere Software wizard reads the options file to determine installation choices. The Update Installer installs the maintenance package in silent mode instead of displaying a graphical user interface.

### Location of the response file

The sample options response file is named install.txt. The file is in the *updi\_root*/responsefiles directory after you install the Update Installer for WebSphere Software into the installation root directory of the WebSphere software product.

## Installing silently

The options file supplies the values to the Update installer wizard when installing silently. The wizard reads the options file to determine responses and does not display the graphical user interface.

The following command uses a copy of the options file named `myresponsefile.txt` to provide installation option responses during a silent installation:

```
./update.sh -options "responsefiles/myresponsefile.txt" -silent
```

If you do not use the `-silent` option, the wizard uses the response file to provide initial values for the graphical interface:

```
./update.sh -options "responsefiles/myresponsefile.txt"
```

## Response file user entry validation

In a silent installation, response file validation is coded into the installation. If the validation does not pass, the failure is recorded in the log files in the `app_server_root/logs/update/tmp` directory.

### Location of the maintenance package to be installed

#### Default directive setting

```
-W maintenance.package=""
```

#### Valid setting

You must set this directive to the location of the maintenance package PAK file. For example, you might specify the following location on a Linux system:

```
/opt/IBM/WebSphere/AppServer/updateinstaller/maintenance/PQ20029.pak
```

#### Error identifiers:

- Maintenance package *maintenance\_package\_name* is already installed on the system.
- Selected product is not supported.
- Configuration failed. The config action that failed was: *configuration\_action*.
- Install the following prerequisite APARs before installing the current maintenance to the target product: *list\_of\_prerequisite\_maintenance\_packages\_to\_install*
- Install the following prerequisite maintenance packages before installing the package you are currently attempting to install: *list\_of\_prerequisite\_maintenance\_packages\_to\_install*
- Uninstall the following APARs before applying the current maintenance to the target product: *list\_of\_prerequisite\_maintenance\_packages\_to\_uninstall*
- Uninstall the following maintenance packages before applying the current maintenance to the target product: *list\_of\_prerequisite\_maintenance\_packages\_to\_uninstall*
- Unable to locate the correct version of the *update\_installer*. Looking for version *version\_identifier*.
- *Maintenance\_package* is not a valid maintenance package.

### Product location

### Default directive setting

-W

product.location="SPECIFY\_PRODUCT\_INSTALL\_LOCATION\_HERE"

### Valid setting

Set this directive to the installation root directory of the product. For example, you might specify the following location on a Linux system:

/opt/IBM/WebSphere/AppServer2

### Error identifiers:

- Maintenance package *maintenance\_package\_name* is already installed on the system.
- Selected product is not supported.
- Configuration failed. The config action that failed was: *configuration\_action*.
- Install the following prerequisite APARs before installing the current maintenance to the target product: *list\_of\_prerequisite\_maintenance\_packages\_to\_install*
- Install the following prerequisite maintenance packages before installing the package you are currently attempting to install: *list\_of\_prerequisite\_maintenance\_packages\_to\_install*
- Uninstall the following APARs before applying the current maintenance to the target product: *list\_of\_prerequisite\_maintenance\_packages\_to\_uninstall*
- Uninstall the following maintenance packages before applying the current maintenance to the target product: *list\_of\_prerequisite\_maintenance\_packages\_to\_uninstall*
- Unable to locate the correct version of *the\_update\_installer*. Looking for version *version\_identifier*.
- *Maintenance\_package* is not a valid maintenance package.
- *Alternate\_product\_directory* could not be validated as an existing directory.

## Usage notes

- The file is not a read-only file.
- Edit this file directly with your flat file editor of choice, such as Kate on SLES or WordPad on a Windows platform.
- The file must exist to perform a silent installation. The Update installer wizard reads this file to determine installation parameters. Provide the fully qualified file path to the backup file.
- Save the copy of the options file in the responsefiles directory for best results.

## Example install.txt file

Edit the version of the file that is included in the Update Installer for WebSphere Software ZIP file. The following example is not guaranteed to be an accurate representation of the actual file.

```
#####  
#  
# This is the silent install response file for installing maintenance packages  
# using the update installer.  
#  
# A common use of an options file is to run the wizard in silent mode. This lets  
# the options file author specify wizard settings without having to run the  
# wizard in graphical or console mode. To use this options file for silent mode  
# execution, *uncomment* and modify the parameters defined within.  
#
```

```

# Use the following command line when running the wizard from the update
# installer directory:
#
#   update -options responsefiles/install.txt -silent
#
# Please enclose all values within a single pair of double quotes.
#
#####

#####
#
# Used to input the maintenance package full filename specification to be installed.
# Edit as appropriate.
#
# ie. -W maintenance.package="C:\Program Files\IBM\WebSphere\AppServer\updateinstaller\maintenance\PQ20029.pak"
#
# Note: If no package is specified, a default of the last downloaded maintenance
# package will be used (based on timestamp).
#
#-W maintenance.package=

#####
#
# Used to input the product install location that will be updated.
#
# ie. -W product.location="C:\Program Files\IBM\WebSphere\AppServer"
#
# Note: The product install location should always been specified, and it should
# always be the full path.
#
-W product.location="<SPECIFY_PRODUCT_INSTALL_LOCATION_HERE>"

#####
#
# Do not edit these values.
#
#
-W update.type="install"

```



---

## Chapter 2. Uninstalling maintenance packages

This topic describes how to use the Update Installer for WebSphere Software to uninstall interim fixes, fix packs, and refresh packs. The Update Installer for WebSphere Software is also known as the update installer program, the updateInstaller program, and the Update installation wizard.

Use the proper authorizations to successfully uninstall product updates. Use the update installer program as the root user on a Linux or UNIX platform, or as the administrator on a Windows platform.

The Update Installer wizard is an InstallShield for Multiplatforms wizard that runs with either a graphical user interface or in silent mode with a response file.

**Important:** See “Known problems and workarounds for the update command” on page 22 for information about known problems and workarounds.

The following descriptions contain reference information about uninstalling interim fixes, fix packs, and refresh packs on WebSphere Application Server products:

### Overview of the uninstall procedure

To uninstall a maintenance package:

1. Use the update installer to install the maintenance package, which creates a backup file in the *app\_server\_root/properties/version/nif/* backup directory. IBM does not support user modifications to backup files.
2. Use the update installer program to remove the maintenance package as described in this topic.

### Viewing the fix level of the node

You can use the Chapter 4, “versionInfo command,” on page 23 in the *app\_server\_root/bin* directory to display the exact fix and version level of the product. However, do not use the versionInfo command while installing or uninstalling a maintenance package.

### Do not launch multiple copies of the Update Installer wizard at one time:

Concurrent launches of the update installer program are not supported. Performing more than one update at the same time can produce unpredictable results, which might include a failed or faulty installation.

### Required information

The graphical interface requires the following information that you must supply:

Table 4. Information required when uninstalling a maintenance package

Field	Valid values	Description
File path of the installation root directory of the WebSphere product and the Update Installer	Identify the installation root directory for one of the following products: <ul style="list-style-type: none"> <li>• IBM WebSphere Application Server</li> <li>• IBM WebSphere Application Server - Express</li> <li>• Embedded version of the IBM WebSphere Application Server - Express</li> <li>• IBM WebSphere Application Server Network Deployment</li> <li>• IBM WebSphere Extended Deployment</li> <li>• IBM Application Client for WebSphere Application Server</li> <li>• IBM WebSphere Business Integration Server Foundation</li> <li>• Web server plug-ins for WebSphere Application Server</li> </ul>	The Update Installer application defaults to the last-visited product location.
File name of the maintenance package to uninstall.	Select a maintenance package to uninstall from the <i>app_server_root/properties/version/update/backup</i> directory.	The default maintenance package is the package with the latest date stamp and time stamp in the <i>app_server_root /properties/version/update/backup</i> directory.

The following procedure describes how to uninstall a maintenance package.

1. Log on to the operating system.

**> Linux** In addition, verify that the umask setting is 022. To verify the umask setting, issue the following command:

```
umask
```

To set the umask setting to 022, issue the following command:

```
umask 022
```

2. Change directories to the *updi\_root* directory.
3. **Windows** Use the Windows Services panel to stop all services for WebSphere Application Server processes.
4. Stop all Java processes that use the IBM Software Developer Kit (SDK) that the WebSphere Application Server product provides.

Before uninstalling interim fixes, fix packs, and refresh packs on a machine, stop all Java processes on the machine that use the IBM SDK, Java Technology Edition that WebSphere Application Server provides.

WebSphere Application Server processes include:

- Application server processes
- The nodeagent process on an application server node when the node is federated into a deployment manager cell
- The dmgr process for the deployment manager server

Stop all Java processes if necessary. If you uninstall a maintenance package while a WebSphere Application Server-related Java process runs, IBM does not guarantee that the product can continue to run successfully, or without error.

5. Use the update installer to uninstall the maintenance package.  
 Uninstall the interim fix on each application server node in a cell before  
 uninstalling the maintenance package from the deployment manager node.  
 Issue one of the following commands to uninstall with the graphical interface:

Table 5. Update installer commands for uninstalling with the graphical interface

Command example	Type of installation	Description
<code>update.bat -W update.type="uninstall"</code>	Graphical interface mode	Initializes the maintenance package field with the name of the maintenance package that was most recently installed.  Accept all of the default values to uninstall the maintenance package with the most recent date stamp and time stamp.
<code>update.bat -W product.location="e:\IBM\WebSphere\AppServer" -W update.type="uninstall"</code>	Graphical interface mode	Overrides the graphical interface with the location of the WebSphere software to update. The default maintenance package to uninstall is the most recently installed maintenance package for that software.
<code>update.bat -W backup.package="PQ20029.pak" -W update.type="uninstall"</code>	Graphical interface mode	Overrides the maintenance package field with the name of the maintenance package to uninstall.
<code>update.bat -W product.location="e:\IBM\WebSphere\AppServer" -W backup.package="PQ20029.pak" -W update.type="uninstall"</code>	Graphical interface mode	Overrides the location of the WebSphere software to update and the name of the maintenance package to uninstall.
<code>update.bat -options "responsefiles/file_name"</code>	Graphical interface mode with an options file	Overrides all default values with values that you specified in the options response file.  If you omit either value from the response file, the default maintenance package is the installed package with the most recent date stamp and time stamp. The default software is the software installed in the parent directory.

Issue the following command to use the silent interface:

Table 6. Update installer command for uninstalling in silent mode

Command example	Type of installation	Description
<code>update.bat -silent -options "responsefiles/file_name"</code>	Silent mode with an options file	Overrides all default values with values that you specified in the options response file.  Always use a response file that is based on the response file under <i>updi_root/responsefiles</i> .

This procedure results in uninstalling maintenance packages to update WebSphere software.

After uninstalling maintenance packages, you can continue to use the WebSphere software.

**Rolling back changes to existing profiles:** Some maintenance packages for WebSphere Application Server products, such as Refresh Pack 2, update existing profiles. If you roll back a maintenance package that contains a profile update, also use any undo scripts provided with the profile update script to roll back changes to the existing profiles.

The readme file for a maintenance package describes scripts that update and scripts that roll back profile fix levels. For example, Refresh Pack 2 for WebSphere Application Server includes required service for the JDBC resource provider templates in existing profiles. See the readme for the profile update and undo scripts for the JDBC-related update for more information.

**Deleting profiles created by a service level that is now rolled back:** See Profiles remain at the Version 6.0.2 level after roll back for a description of a limitation that requires profiles to be at the same service level or at a lower service level than the WebSphere Application Server product.

For example, suppose that you install Fix Pack 1 for Version 6.1 (Version 6.1.0.1), create a new profile, and then roll back Fix Pack 1. You must delete the profile that you created at the Version 6.1.0.1 level to avoid possible problems.

---

## uninstall.txt

The Update Installer for WebSphere Software can use an options response file to uninstall maintenance packages from a command line interface.

The `uninstall.txt` file has one directive that identifies the backup file for uninstalling a service update. Comments in the file describe how to set the string value.

The Update Installer for WebSphere Software wizard reads the options file to determine uninstall choices. The Update Installer uninstalls the maintenance package in silent mode, instead of displaying a graphical user interface.

### Location of the response file

The sample options response file is named `uninstall.txt`. The file is in the `updi_root/responsefiles` directory after you unzip the Update Installer for WebSphere Software into the installation root directory of the WebSphere software product.

### Uninstalling silently

The options file supplies the values to the Update installer wizard when uninstalling silently. The wizard reads the options file to determine responses and does not display the graphical user interface.

The following command uses a copy of the options file named `myresponsefile.txt` to provide uninstall option responses during a silent uninstall:

```
./update.sh -options "responsefiles/myresponsefile.txt" -silent
```

If you do not use the `-silent` option, the wizard uses the response file to provide initial values for the graphical interface:

```
./update.sh -options "responsefiles/myresponsefile.txt"
```

## Response file user entry validation

In a silent uninstall, response file validation has been coded into the uninstall process. If the validation does not pass, the failure is recorded in the log files in the `app_server_root/logs/update/tmp` directory.

### Location of the maintenance package to be uninstalled

#### Default directive setting

```
-W backup.package=""
```

#### Valid setting

You must set this directive to the location of the backup file. The backup file reverses the application of the maintenance. For example, you might specify the following location on a Linux system:

```
/opt/properties/version/nif/backup/maintenance_package_to_uninstall
```

#### Error identifiers:

- The maintenance package cannot be uninstalled. Uninstalling the maintenance would break the following superseding maintenance packages. Uninstall the superseding maintenance packages first: *list\_of\_superseding\_maintenance\_packages*
- This maintenance package cannot be uninstalled. The following maintenance packages are dependent on the package that you are attempting to uninstall: *list\_of\_dependent\_maintenance\_packages*
- This maintenance package cannot be uninstalled. The following maintenance packages are dependent on the APARs you are attempting to uninstall: *list\_of\_dependent\_maintenance\_packages*
- No installation backup packages are available for uninstalling maintenance.

### Product location

Although uninstalling maintenance from another product is possible, always use the Update installer wizard from the directory structure of the product that you are updating if possible. Problems can occur when a mismatch between product SDKs occurs, for example.

Do not use this directive unless absolutely necessary.

#### Default directive setting

```
-W  
product.location="SPECIFY_PRODUCT_INSTALL_LOCATION_HERE"
```

#### Valid setting

You must set this directive to the installation root directory of the alternate product. For example, you might specify the following location on a Linux system:

```
/opt/IBM/WebSphere/AppServer2
```

#### Error identifiers:

- The maintenance package cannot be uninstalled. Uninstalling the maintenance would break the following superseding maintenance packages. Uninstall the superseding maintenance packages first: *list\_of\_superseding\_maintenance\_packages*
- This maintenance package cannot be uninstalled. The following maintenance packages are dependent on the package that you are attempting to uninstall: *list\_of\_dependent\_maintenance\_packages*

- This maintenance package cannot be uninstalled. The following maintenance packages are dependent on the APARs you are attempting to uninstall: *list\_of\_dependent\_maintenance\_packages*
- No installation backup packages are available for uninstalling maintenance.
- 

## Usage notes

- The file is not a read-only file.
- Edit this file directly with your flat file editor of choice, such as Kate on SLES or WordPad on a Windows platform.
- The file must exist to perform a silent uninstall. The Update installer wizard reads this file to determine uninstall parameters. Provide the fully qualified file path to the backup file.
- Save the copy of the options file in the responsefiles directory for best results.

## Example uninstall.txt file

Edit the version of the file that is included in the Update Installer for WebSphere Software ZIP file. The following example is not guaranteed to be an accurate representation of the actual file.

```
#####
#
# This is the silent install response file for uninstalling maintenance packages
# using the update installer.
#
# A common use of an options file is to run the wizard in silent mode. This lets
# the options file author specify wizard settings without having to run the
# wizard in graphical or console mode. To use this options file for silent mode
# execution, *uncomment* and modify the parameters defined within.
#
# Use the following command line when running the wizard from the update
# installer directory:
#
#   update -options responsefiles/uninstall.txt -silent
#
# Please enclose all values within a single pair of double quotes.
#####

#####
#
# Used to input the maintenance backup package filename to be uninstalled.
# This is the same filename as the package that was originally installed.
# A maintenance package can only be uninstalled if a backup package exists.
#
# ie. -W backup.package="PQ20029.pak"
#
# Note: If no package is specified, a default of the last installed maintenance
# package will be used.
#
#-W backup.package=""

#####
#
# Used to modify the product install location that will be updated.
# This value should be left commented out if the Update Installer is
# being run from the recommended location
#
# ie. -W product.location="C:\Program Files\IBM\WebSphere\AppServer"
```

```
#
# Note: The product install location should always been specified, and it should
# always be the full path.
#
-W product.location="<SPECIFY_PRODUCT_INSTALL_LOCATION_HERE>"

#####
#
# Do not edit these values.
#
-W update.type="uninstall"
```



---

## Chapter 3. update command

The **update** command is the Update Installer for WebSphere Software program. The Update installer wizard is also known as the Update installation wizard, the update installer program, and the updateInstaller program.

The update installer program installs and uninstalls interim fixes, fix packs, and refresh packs to update WebSphere software.

### Overview

The **update** command calls the update installer program to install and uninstall service to update WebSphere software. This topic describes the update installer command and its command-line parameters.

The following descriptions contain reference information about the command.

See Chapter 1, "Installing maintenance packages," on page 1 and Chapter 2, "Uninstalling maintenance packages," on page 11 for information about using the command.

**Important:** See "Known problems and workarounds for the update command" on page 22 for information about known problems and workarounds.

---

## Command options

The following tables list commands for installing and uninstalling interim fixes.

### Commands for installing interim fixes

Issue one of the following commands to use the graphical interface:

Table 7. Update installer commands for installing with the graphical interface

Command example	Type of installation	Description
update.bat	Graphical interface mode	Initializes the interim fix field with the name of the interim fix that has the most recent date stamp and time stamp.  Accept all of the default values to install the interim fix with the most recent time stamp.
update.bat -options "responsefiles/file_name"	Graphical interface mode with an options file	Overrides all graphical interface values with values that you specified in the options response file.  If you omit either value, the default maintenance package is the one with the most recent date stamp and time stamp. The default software is the software installed in the parent directory.

Table 7. Update installer commands for installing with the graphical interface (continued)

Command example	Type of installation	Description
update.bat -W maintenance.package="e: \IBM\WebSphere\AppServer\updateinstaller\ maintenance\PQ20029.pak"	Graphical interface mode	Overrides the name of the maintenance package to apply.
update.bat -W product.location="e: \IBM\WebSphere\AppServer"	Graphical interface mode	Overrides the location of the WebSphere software to update.
update.bat -W product.location="e: \IBM\WebSphere\AppServer" -W maintenance.package="e: \IBM\WebSphere\ AppServer\updateinstaller\maintenance \PQ20029.pak"	Graphical interface mode	Overrides the location of the WebSphere software to update and the name of the maintenance package to apply.

Issue the following command to use the silent interface:

Table 8. Update installer command for installing in silent mode

Command example	Type of installation	Description
update.bat -silent -options "responsefiles/ <i>file_name</i> "	Silent mode with an options file	Overrides all default values with values that you specified in the options response file.  Always use a response file that is based on the response file under <i>updi_root</i> /responsefiles.

## Commands for uninstalling interim fixes

Issue one of the following commands to uninstall with the graphical interface:

Table 9. Update installer commands for uninstalling with the graphical interface

Command example	Type of installation	Description
update.bat -W update.type="uninstall"	Graphical interface mode	Initializes the interim fix field with the name of the interim fix that was most recently installed.  Accept all of the default values to uninstall the interim fix with the most recent date stamp and time stamp.
update.bat -W product.location="e: \IBM\WebSphere\AppServer" -W update.type="uninstall"	Graphical interface mode	Overrides the graphical interface with the location of the WebSphere software to update. The default interim fix to uninstall is the most recently installed interim fix for that software.
update.bat -W backup.package="PQ20029.pak" -W update.type="uninstall"	Graphical interface mode	Overrides the interim fix field with the name of the maintenance package to uninstall.
update.bat -W product.location="e: \IBM\WebSphere\AppServer" -W backup.package="PQ20029.pak" -W update.type="uninstall"	Graphical interface mode	Overrides the location of the WebSphere software to update and the name of the maintenance package to uninstall.

Table 9. Update installer commands for uninstalling with the graphical interface (continued)

Command example	Type of installation	Description
update.bat -options "responsefiles/file_name"	Graphical interface mode with an options file	<p>Overrides all default values with values that you specified in the options response file.</p> <p>If you omit either value from the response file, the default maintenance package is the installed package with the most recent date stamp and time stamp. The default software is the software installed in the parent directory.</p>

Issue the following command to use the silent interface:

Table 10. Update installer command for installing in silent mode

Command example	Type of installation	Description
update.bat -silent -options "responsefiles/file_name"	Silent mode with an options file	<p>Overrides all default values with values that you specified in the options response file.</p> <p>Always use a response file that is based on the response file under <i>updi_root/responsefiles</i>.</p>

## Logging

The following sections describe logging that occurs when installing and uninstalling service.

### Logs created when installing service

If no installation log file exists, refer to the temporary log file in the *updi\_root/logs/update/tmp* directory. If all validations pass, the installation occurs.

Then the update installer program creates the *app\_server\_root/logs/update/maintenance\_package.install* directory.

Within the directory are the *update.log.txt* file, the compressed *update.trace.log.gz* file, and the compressed *update.config.log.gz* file. The *update.config.log.gz* file exists only when the installation of service uses the internal configuration manager utility to run ANT scripts.

### Logs created when uninstalling service

If no log file exists after uninstalling an interim fix, refer to the temporary log file in the *updi\_root/logs/update/tmp* directory. If all validations pass, the uninstall procedure occurs.

Then the update installer program creates the *app\_server\_root/logs/update/maintenance\_package.uninstall* directory.

Within the directory are the `updatelog.txt` file, the compressed `updatetrace.log.gz` file, and the compressed `updateconfig.log.gz` file. The `updateconfig.log.gz` file exists only when the removal of service uses the internal configuration manager utility to run ANT scripts.

## Indicators of success

The log file includes an indicator of success:

### **INSTCONFSUCCESS**

The current operation was successful. You do not need to review the log file any further.

### **INSTCONFPARTIALSUCCESS**

The current operation was partially successful. System should still be in a usable state, however some non-critical actions have failed. Consult the log file to determine what has failed and how to recover from the failure, if possible.

### **INSTCONFFAILED**

The current operation failed. The system is no longer in a usable state. Consult the log file for more information.

---

## Known problems and workarounds for the update command

This topic describes known problems and issues associated with the Update Installer for WebSphere Software program.

The update installer program displays its version information in the title bar of the graphical interface. Version information is stored in the `version.txt` file in the `updateinstaller` directory.

A new version might ship to correspond to any new fix. Information in the `version.txt` file is displayed prominently in the title bar of the wizard and is also recorded in the `updatelog.txt` file.

Always download and use the latest version of the Update installer wizard when installing an interim fix.

The Update Installer can not automatically detect locks on files by remote processes. So you must ensure that all AppServers processes have been stopped for all your profiles, including any remote profiles.

---

## Chapter 4. versionInfo command

The **versionInfo** command generates a report from data extracted from XML files in the properties/version folder. The report includes a list of changed components and installed or uninstalled maintenance packages.

### Product version information

The versionInfo tool displays important data about the product and its installed components, such as the build version and build date. History information for installation and removal of maintenance packages also displays in the report. This tool is particularly useful when working with support personnel to determine the cause of any problem.

### Product version reports

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

- versionInfo script  
Lets you use parameters to create a version report.
- genVersionReport script  
Generates the versionReport.html report file in the current working directory, which is usually the bin directory.

---

## Location of the command file

The versionInfo command is a script.

**Linux** The command file is a script named versionInfo.sh in the *app\_server\_root/bin* directory.

**Windows** The command file is named versionInfo.bat in the *app\_server\_root\bin* directory.

---

## Syntax for the versionInfo command

**Linux**

The command syntax is:

```
versionInfo.sh [ -format text | html ]  
               [ -file file_name ]  
               [ -long ]  
               [ -maintenancePackages ]  
               [ -maintenancePackageDetail ]  
               [ -components ]  
               [ -componentDetail ]
```

```
versionInfo [ -help | /help | -? | /? | -usage ]
```

Issue the command from the *app\_server\_root/bin* directory.

**Windows**

The command syntax is:

```
versionInfo [ -format text | html ]  
            [ -file file_name ]  
            [ -long ]  
            [ -maintenancePackages ]  
            [ -maintenancePackageDetail ]  
            [ -components ]  
            [ -componentDetail ]
```

```
versionInfo [ -help | /help | -? | /? | -usage ]
```

Issue the command from the *app\_server\_root*\bin directory.

---

## Parameters

**-? or /?**

Displays command syntax.

**-components**

Adds a list of installed components to the report.

**-componentDetail**

Adds details about installed components to the report.

**-file *file\_name***

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

**-format text | html**

Selects the format of the report. The default is "text".

**-help or /help**

Displays command syntax.

**-long**

Creates the long version of the report.

**-maintenancePackages**

Adds a list of applied maintenance packages to the report.

**-maintenancePackageDetail**

Adds details about an applied maintenance package to the report.

**-usage**

Displays command syntax.

---

## Report description

The versionInfo command reports the following information:

### Installation information

Displays the following general information about the current installation:

- Report date and time - The date and time that the report was generated. The timestamp is formatted according to the current locale.
- Product directory - The file path to the installation root directory defined by the WAS\_HOME environment variable.
- Version directory - The file path of the version directory of the current IBM WebSphere Application Server - ND installation.

- DTD directory - The file path of the DTD directory of the current IBM WebSphere Application Server - ND installation.
- Log directory - The file path of the log directory of the current IBM WebSphere Application Server - ND installation. The maintenance package log files are in the directory.
- Backup directory - The file path of the backup directory of the current IBM WebSphere Application Server - ND installation. The backup files generated during the installation of maintenance packages are in this directory.
- TMP directory - The file path of the temporary directory of the current machine.

## Product list information

Displays a list of installed WebSphere products:

- Product ID - The product ID of the installed product.
- Status - The status of the product, either installed or uninstalled.

## Installed product information

This information and the other information topic descriptions are hierarchal for each installed product, component, component update, installed maintenance package, included APARs, and component updates.

This section of the report displays the following information:

- Name - The name of the installed product.
- Version - The current version of the product. Installing or uninstalling fix packs or refresh packs modifies this version.
- ID - The product ID of the product installed, such as BASE, BASETRIAL, ND, EXPRESS, EXPRESSTRIAL, embeddedEXPRESS, IHS, XD, PLG, or CLIENT.
- Build level - The build level of the installed product.
- Build date - The build date of the installed product.

## Installed component information

Displays the following component-level information of the installed component from the .component file under the /properties/version directory:

- Component name - The name of the installed component
- Spec version - The spec version of the current component
- Build level - The build level of the current component
- Build date - The build date of the current component

**Installed component update information:** Displays the general maintenance package information:

- Component name - the name of the installed component
- Update type - All interim fixes, fix packs and refresh packs are maintenance packages. The update type field is always set to *maintenance package*.
- Maintenance package ID - The ID of the maintenance package that is responsible for the current maintenance level of the product.
- Update effect - The updated action taken on a particular component. The default effect is *replace*.
- Log file name - The file path of the log file that records the maintenance actions for the current maintenance level.

- Backup file name - The file path of the backup file generated during the installation of the maintenance package. Not applicable if the component is restored as a result of an uninstall of a maintenance package.
- Timestamp - The time at which the component is updated. The timestamp is stated in GMT offset values.

*Installed maintenance package information:* Displays the general maintenance package information:

- Maintenance Package ID - the maintenance package ID
- Description - the description of the maintenance package
- Build Date - the build date of this maintenance package

*Included APARs information:* Displays the list of APARs fixed by this maintenance package.

*Component updates information:* Displays the following information about each component that is updated by the installed maintenance package:

- Component name - The name of the installed component.
- Updated effect - The update action taken on the component by the maintenance package. The default action is *replace*.
- Timestamp - The time at which the component update is installed (GMT offset).

---

## Sample versionInfo report

When the WebSphere Application Server product has no interim fixes or fix packs applied, the genVersionReport.bat script creates the following information in the versionReport.html report file, which is edited to show only the first few components:

IBM WebSphere Application Server Product Installation Status Report

```

-----
Report at date and time 2005-05-18 15:58:40-0400

Installation
-----
Product Directory: /opt/WebSphere/AppServer
Version Directory: /opt/WebSphere/AppServer/properties/version
DTD Directory:    /opt/WebSphere/AppServer/properties/version/dtd
Log Directory:   /opt/WebSphere/AppServer/logs/update
Backup Directory: /opt/WebSphere/AppServer/properties/version/backup
TMP Directory:   /tmp

Installation Platform
-----
Name           IBM WebSphere Application Server
Version        6.0

Product List
-----
BASE           installed

Installed Product
-----
Name           IBM WebSphere Application Server
Version        6.0.1

```

ID                    BASE  
Build Level         m0451.03  
Build Date         12/19/2004

Installed Component

-----  
Component Name:        activity.impl  
Spec Version:         6.0  
Build Version:        m0451.03  
Build Date:           12/19/04

Installed Component Update

-----  
Component Name:        activity.impl  
Update Type:           maintenance package  
Maintenance Package ID: was60\_fp1\_linux  
Update Effect:         replace  
Log File Name:  
  /opt/WebSphere/AppServer/logs/update/was60\_fp1\_linux.install/updatelog.txt  
Backup File Name:  
  /opt/WebSphere/AppServer/properties/version/backup/was60\_fp1\_linux.pak  
Timestamp:            2004-12-17 18:24:34-0500

Installed Component Update

-----  
Component Name:        activity.impl  
Update Type:           maintenance package  
Maintenance Package ID: was60\_fp2  
Update Effect:         replace  
Log File Name:  
  /opt/WebSphere/AppServer/logs/update/was60\_fp2\_linux.install/updatelog.txt  
Backup File Name:  
  /opt/WebSphere/AppServer/properties/version/backup/was60\_fp2\_linux.pak  
Timestamp:            2004-12-19 06:24:34-0500

Installed Component

-----  
Component Name:        activity.session  
Spec Version:         6.0  
Build Level:         m0451.03  
Build Date:         12/19/04

Installed Component Update

-----  
Component Name:        activity.session  
Update Type:           maintenance package  
Maintenance Package ID: was60\_fp2  
Update Effect:         replace  
Log File Name:  
  /opt/WebSphere/AppServer/logs/update/was60\_fp2\_linux.install/updatelog.txt  
Backup File Name:  
  /opt/WebSphere/AppServer/properties/version/backup/was60\_fp2\_linux.pak  
Timestamp:            2004-12-19 06:24:34-0500

Installed Maintenance Package

-----  
Maintenance Package ID: was60\_fp1\_linux  
Description:           IBM WebSphere Application Server,  
                      Version 6.0.1 Fix Pack for Linux  
Build Date:            12/17/2004

Included Apars

-----  
PQ12345

```

Component Updates
-----
activity      updated
installed on 2004-12-17 06:24:30-0500

Installed Maintenance Package
-----
Maintenance Package ID:  was60_fp2
Description:             IBM WebSphere Application Server,
                        Version 6.0.2 Fix Pack for Linux
Build Date:              12/19/2004

Included Apars
-----
PQ12345
PQ23456
PQ34567

Component Updates
-----
activity      updated
installed on 2004-12-19 06:24:30-0500
activity.impl updated
installed on 2004-12-19 06:24:30-0500

-----
End Installation Status Report
-----

```

---

## Summary of Version 6 changes for the versionInfo command

Changes are in two areas: command syntax and report information.

### Changes to command syntax

The following changes are in effect:

- Version 6 replaces the terms *efixes* and *ptfs* with *maintenancePackages* to describe a specific maintenance package. This matches the terminology used in the Version 6 update installer application.
- Version 6 replaces the terms *efixesDetail* and *ptfDetail* with *maintenancePackageDetail* to describe the detailed information that relates to a specific maintenance package.

### Changes to report information

The following changes are in effect:

- Version 6 replaces the term *technology list* with *product list*. The technology list displays the list of installed products in the current WAS\_HOME directory.
- Version 6 uses an update type value of *maintenance package*.
- Version 6 replaces the term *PTF ID* with *Maintenance Package ID*. This change is consistent with the terminology used in the Update Installer for WebSphere Software.
- Version 6 removes *IsRequired*, *IsOptional*, *IsExternal*, and *IsCustom* from the *Installed component update* section of the report. The older terminology is not supported per component by the Version 6 Update Installer for WebSphere Software.
- Version 6 replaces the term *Install Date* with *Timestamp* for consistency with the History report.

- Version 6 replaces the *Installed PTF* section of the report with the *Installed Maintenance Package* section.
- Version 6 removes the term *Exposition* because it is similar to the term *Description*.
- Version 6 removes the *Build Version* field from the *Installed Maintenance Package* section.
- Version 6 removes the *Supported Platforms* section of the report.
- Version 6 replaces the *Included Fixes* section of the report with the with by *Included APARs* section, which lists the APARs.
- Version 6 removes the *Custom Properties* section of the report.



---

## Chapter 5. historyInfo command

The **historyInfo** command generates a report from data extracted from XML files in the properties/version folder and the properties/version/history folder. The report includes a list of changed components and a history of installed or uninstalled maintenance packages.

### Product history information

The historyInfo tool displays important data about the product and its installed components, such as the build version and build date. History information for installation and removal of maintenance packages also displays in the report. This tool is particularly useful when working with support personnel to determine the cause of any problem.

### Product history reports

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

- historyInfo script  
Lets you use parameters to create a history report.
- genHistoryReport script  
Generates the historyReport.html report file in the current working directory, which is usually the bin directory.

---

## Location of the command file

The historyInfo command is a script.

**Linux** The command file is a script named genHistoryReport.sh in the *app\_server\_root/bin* directory.

**Windows** The command file is a script named genHistoryReport.bat in the *app\_server\_root\bin* directory.

---

## Syntax for the historyInfo command

**Linux**

The command syntax is:

```
historyInfo.sh [ -format text | html ]  
               [ -file file_name ]  
               [ -maintenancePackageID ID_of_maintenance_package ]  
               [ -component component_name ]
```

```
historyInfo [ -help | /help | -? | /? | -usage ]
```

Issue the command from the *app\_server\_root/bin* directory.

**Windows**

The command syntax is:

```
historyInfo [ -format text | html ]  
            [ -file file_name ]  
            [ -maintenancePackageID ID_of_maintenance_package ]  
            [ -component component_name ]
```

```
historyInfo [ -help | -? | /help | /? | -usage ]
```

Issue the command from the *app\_server\_root*\bin directory.

---

## Parameters

### **-? or /?**

Displays command syntax.

### **-component** *component\_name*

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.

### **-file** *file\_name*

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

### **-format text | html**

Selects the format of the report. The default is "text".

### **-help or /help**

Displays command syntax.

### **-maintenancePackageID** *ID\_of\_maintenance\_package*

Specifies the ID of the interim fix, fix pack, or refresh pack. When specified, the product history report displays events for only the named maintenance package. When not specified, the report displays events for all maintenance packages.

### **-usage**

Displays command syntax.

---

## Report description

The historyInfo command reports the following information:

### Installation information

Displays the following general information about the current installation:

- Report date and time - The date and time that the report was generated. The timestamp is formatted according to the current locale.
- Product directory - The file path to the installation root directory of the product.
- Version directory - The file path of the version directory of the current WAS installation.
- DTD directory - The file path of the DTD directory of the current installation.
- Log directory - The file path of the log directory of the current installation. The maintenance package log files are in the directory.
- Backup directory - The file path of the backup directory of the current WAS installation. The backup files generated during the installation of maintenance packages are in this directory.

- TMP directory - The file path of the temporary directory of the current machine.
- History directory - The file path of the history directory of the current installation. The history files are in the directory.
- History File - The file path of the event.history file.

## Installation event information

Displays the list of installed maintenance packages (interim fix, fix pack, and refresh pack) and the following related information:

- Maintenance package ID - The ID of the maintenance package.
- Action - The action taken with this maintenance package, which is either *install* or *uninstall*.
- Package file name - The file name of the maintenance package that was installed.
- Log file name - The file path of the log file generated during the installation or removal of the maintenance package.
- Backup file name - The file path of the backup file generated during the installation of the maintenance package. This field does not apply for an uninstall action.
- Timestamp - The time when the maintenance action (install or uninstall) occurs. The time is stated in relation to GMT.
- Result - The result of the installation or removal action. The result is either success, partial success, or failure.

## Component installation event information

Displays the following component-level information of the event for the current maintenance package:

- Maintenance package ID - The ID of the maintenance package to which this particular installation event belongs.
- Component name - The name of the current component.
- Action - The action taken on this component due to the action of the current maintenance package, either install or uninstall
- Update action - The updated action taken on this component. By default, the update action for an installation action is *replace*.
- Timestamp - The time at which the action occurs for the maintenance package (GMT offset values).
- Result - The result of the install or uninstall action. The result is either success, partial success or failure.

---

## The event.history file

The historyInfo command also generates the event.history file. This file represents the raw data of the history report information. The following example of an event.history file corresponds to the history report in the preceding example:

```
<!DOCTYPE event-history SYSTEM "eventHistory.dtd">
<event-history>
  <update-event
    event-type="ptf"
    id="was60_fp1_linux"
    update-action="install"
    primary-content="was60_fp1_linux.pak"
    update-type="replace"
    log-name=
```

```

    "/opt/WebSphere/AppServer/logs/update/was60_fp1_linux.install/updatelog.txt"
backup-name=
    "/opt/WebSphere/AppServer/properties/version/backup/was60_fp1_linux.pak"
start-time-stamp="2004-12-14 06:15:14-0500"
result="success">
<update-event
    event-type="component"
    parent-id="was60_fp1_linux"
    id="activity"
    update-action="install"
    update-type="replace"
    start-time-stamp="2004-12-14 06:15:14-0500"
    result="success">
</update-event>
<update-event
    event-type="component"
    parent-id="was60_fp1_linux"
    id=" activity.impl"
    update-action="install"
    update-type="replace"
    start-time-stamp="2004-12-14 06:15:14-0500"
    result="success">
</update-event>
</update-event>
<update-event
    event-type="ptf"
    id="was60_fp2"
    update-action="install"
    primary-content="was60_fp1_linux.pak"
    update-type="replace"
    log-name="/opt/WebSphere/AppServer/logs/update/was60_fp2.install/updatelog.txt"
    backup-name="/opt/WebSphere/AppServer/properties/version/backup/was60_fp2.pak"
    start-time-stamp="2004-12-14 10:25:34-0500"
    result="partialSuccess">
<update-event
    event-type="component"
    parent-id="was60_fp2"
    id="activity"
    update-action="install"
    update-type="replace"
    start-time-stamp="2004-12-14 10:25:34-0500"
    result="partialSuccess">
</update-event>
<update-event
    event-type="component"
    parent-id="was60_fp2"
    id=" activity.impl"
    update-action="install"
    update-type="replace"
    start-time-stamp="2004-12-14 10:25:34-0500"
    result="partialSuccess">
</update-event>
</update-event>
<update-event
    event-type="ptf"
    id="was60_fp2"
    update-action="uninstall"
    primary-content=" was60_fp2.pak"
    update-type="replace"
    log-name=
        "/opt/WebSphere/AppServer/logs/update/was60_fp2.uninstall/updatelog.txt"
    backup-name="not applicable"
    start-time-stamp="2004-12-18 17:29:12-0500"
    result="partialSuccess">
<update-event
    event-type="component"
    parent-id="was60_fp2"

```

```

        id="activity"
        update-action="uninstall"
        update-type="replace"
        start-time-stamp="2004-12-18 17:29:12-0500"
        result="partialSuccess">
</update-event>
<update-event
  event-type="component"
  parent-id="was60_fp2"
  id=" activity.impl"
  update-action="uninstall"
  update-type="replace"
  start-time-stamp="2004-12-18 17:29:12-0500"
  result="partialSuccess">
</update-event>
</update-event>
<update-event
  event-type="ptf"
  id="was60_fp1_linux"
  update-action="uninstall"
  primary-content=" was60_fp1_linux.pak"
  update-type="replace"
  log-name=
    "/opt/WebSphere/AppServer/logs/update/was60_fp1_linux.install/update.log.txt"
  backup-name="not applicable"
  start-time-stamp="2004-12-23 15:15:14-0500"
  result="failure">
<update-event
  event-type="component"
  parent-id="was60_fp1_linux"
  id="activity"
  update-action="uninstall"
  update-type="replace"
  start-time-stamp="2004-12-23 15:15:14-0500"
  result="failure">
</update-event>
<update-event
  event-type="component"
  parent-id="was60_fp1_linux"
  id=" activity.impl"
  update-action="uninstall"
  update-type="replace"
  start-time-stamp="2004-12-23 15:15:14-0500"
  result="failure">
</update-event>
</update-event>
</event-history>

```

---

## Summary of Version 6 changes for the historyInfo command

Changes are in three areas: command syntax, report information, and the event.history file.

### Changes to command syntax

Version 6 replaces the term *updateID* with *maintenancePackageID* to describe a specific maintenance package. This matches the terminology used in the Version 6 update installer application.

### Changes to report information

The following changes are in effect:

- Version 6 replaces the term *PTF ID* with *Maintenance package ID*. This change is consistent with the terminology used in the Update Installer for WebSphere Software.
- Version 6 replaces the term *Primary content* with *Package file name*.
- Version 6 replaces the terms *Start Time* and *End Time* with a *Timestamp* for the completion of the event.
- Version 6 removes the terms *IsExternal*, and *IsCustom*.
- Version 6 removes the *Result Message* section of the report because the information is reported in the *Result and Action* section.
- Version 6 sets the value of *Backup file name* to "not applicable" if the update action is *uninstall*. No backup file is generated when a maintenance package is uninstalled.
- Version 6 adds a new result state, partial success. The Version 6 Update Installer for WebSphere Software can generate a partial success.

## Changes to the event.history file

The following changes are in effect:

- Version 6 replaces the terms *Start Time* and *End Time* with a *Timestamp* for the completion of the event.
- Version 6 removes the terms *IsRequired*, *IsExternal*, *IsCustom*, *root-property-file*, *root-property-name*, and *root-property-value*, which are not supported by the Update Installer for WebSphere Software.
- Version 6 removes the *Result Message* section of the report because the information is reported in the *Result and Action* section.
- Version 6 sets the value of *update-event* to "replace".
- Version 6 sets the value of *Backup file name* to "not applicable" if the update action is *uninstall*. No backup file is generated when a maintenance package is uninstalled.
- Version 6 adds a new result state, partial success. The Version 6 Update Installer for WebSphere Software can generate a partial success.

---

## Chapter 6. genHistoryReport command

The **genHistoryReport** command generates the `historyReport.html` report file in the current working directory, which is usually the `bin` directory. The report includes a list of changed components and installed or uninstalled maintenance packages. The `genHistoryReport` script invokes the `historyInfo` script specifying the correct parameters to place the information generated into an HTML file in the current directory.

### Product history information

The `historyInfo` tool displays historical data about the product and the installation and removal of maintenance packages for the product. This tool is particularly useful when working with support personnel to determine the cause of any problem.

### Product history reports

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

- `historyInfo` script  
Lets you use parameters to create a history report.
- `genHistoryReport` script  
Generates the `historyReport.html` report file in the current working directory, which is usually the `bin` directory. The report includes a list of components, fixes, fix packs, and refresh packs.

---

## Location of the command file

The command file is a script.

**Linux** The command file is named `genHistoryReport.sh` in the `app_server_root/bin` directory.

**Windows** The command file is named `genHistoryReport.bat` in the `app_server_root\bin` directory.

---

## Syntax for the genHistoryReport command

**Linux**

The command syntax is:

```
genHistoryReport.sh
```

Issue the command from the `app_server_root/bin` directory.

**Windows**

The command syntax is:

```
genHistoryReport.bat
```

Issue the command from the `app_server_root\bin` directory.

---

## Report description

The `historyInfo` command generates the report. The `genHistoryReport` command calls the `historyInfo` command with a set of report parameters that reports the following information:

### Installation information

Installation information displays the following general information about the current installation:

- Report date and time - The date and time that the report was generated. The timestamp is formatted according to the current locale.
- Product directory - The file path to the installation root directory of the product.
- Version directory - The file path of the version directory of the current WAS installation.
- DTD directory - The file path of the DTD directory of the current installation.
- Log directory - The file path of the log directory of the current installation. The maintenance package log files are in the directory.
- Backup directory - The file path of the backup directory of the current WAS installation. The backup files generated during the installation of maintenance packages are in this directory.
- TMP directory - The file path of the temporary directory of the current machine.
- History directory - The file path of the history directory of the current installation. The history files are in the directory.
- History File - The file path of the `event.history` file.

### Installation event information

Installation event information displays the list of installed maintenance packages (interim fix, fix pack, and refresh pack) and the following related information:

- Maintenance package ID - The ID of the maintenance package.
- Action - The action taken with this maintenance package, which is either *install* or *uninstall*.
- Package file name - The file name of the maintenance package that was installed.
- Log file name - The file path of the log file generated during the installation or removal of the maintenance package.
- Backup file name - The file path of the backup file generated during the installation of the maintenance package. This field does not apply for an uninstall action.
- Timestamp - The time when the maintenance action (install or uninstall) occurs. The time is stated in relation to GMT.
- Result - The result of the installation or removal action. The result is either success, partial success, or failure.

### Component installation event information

Component installation event information displays the following component-level information of the event for the current maintenance package:

- Maintenance package ID - The ID of the maintenance package to which this particular installation event belongs.

- Component name - The name of the current component.
- Action - The action taken on this component due to the action of the current maintenance package, either install or uninstall
- Update action - The updated action taken on this component. By default, the update action for an installation action is *replace*.
- Timestamp - The time at which the action occurs for the maintenance package (GMT offset values).
- Result - The result of the install or uninstall action. The result is either success, partial success or failure.

---

## The event.history file

The genHistoryReport command also generates the event.history file. This file represents the raw data of the history report information. The following example of an event.history file corresponds to the history report in the preceding example:

```
<!DOCTYPE event-history SYSTEM "eventHistory.dtd">
<event-history>
  <update-event
    event-type="ptf"
    id="was60_fp1_linux"
    update-action="install"
    primary-content="was60_fp1_linux.pak"
    update-type="replace"
    log-name=
      "/opt/WebSphere/AppServer/logs/update/was60_fp1_linux.install/updatelog.txt"
    backup-name=
      "/opt/WebSphere/AppServer/properties/version/backup/was60_fp1_linux.pak"
    start-time-stamp="2004-12-14 06:15:14-0500"
    result="success">
    <update-event
      event-type="component"
      parent-id="was60_fp1_linux"
      id="activity"
      update-action="install"
      update-type="replace"
      start-time-stamp="2004-12-14 06:15:14-0500"
      result="success">
    </update-event>
    <update-event
      event-type="component"
      parent-id="was60_fp1_linux"
      id=" activity.impl"
      update-action="install"
      update-type="replace"
      start-time-stamp="2004-12-14 06:15:14-0500"
      result="success">
    </update-event>
  </update-event>
  <update-event
    event-type="ptf"
    id="was60_fp2"
    update-action="install"
    primary-content="was60_fp1_linux.pak"
    update-type="replace"
    log-name="/opt/WebSphere/AppServer/logs/update/was60_fp2.install/updatelog.txt"
    backup-name="/opt/WebSphere/AppServer/properties/version/backup/was60_fp2.pak"
    start-time-stamp="2004-12-14 10:25:34-0500"
    result="partialSuccess">
    <update-event
      event-type="component"
      parent-id="was60_fp2"
      id="activity"
      update-action="install"
```

```

        update-type="replace"
        start-time-stamp="2004-12-14 10:25:34-0500"
        result="partialSuccess">
</update-event>
<update-event
  event-type="component"
  parent-id="was60_fp2"
  id=" activity.impl"
  update-action="install"
  update-type="replace"
  start-time-stamp="2004-12-14 10:25:34-0500"
  result="partialSuccess">
</update-event>
</update-event>
<update-event
  event-type="ptf"
  id="was60_fp2"
  update-action="uninstall"
  primary-content=" was60_fp2.pak"
  update-type="replace"
  log-name=
    "/opt/WebSphere/AppServer/logs/update/was60_fp2.uninstall/updatelog.txt"
  backup-name="not applicable"
  start-time-stamp="2004-12-18 17:29:12-0500"
  result="partialSuccess">
  <update-event
    event-type="component"
    parent-id="was60_fp2"
    id="activity"
    update-action="uninstall"
    update-type="replace"
    start-time-stamp="2004-12-18 17:29:12-0500"
    result="partialSuccess">
  </update-event>
  <update-event
    event-type="component"
    parent-id="was60_fp2"
    id=" activity.impl"
    update-action="uninstall"
    update-type="replace"
    start-time-stamp="2004-12-18 17:29:12-0500"
    result="partialSuccess">
  </update-event>
</update-event>
<update-event
  event-type="ptf"
  id="was60_fp1_linux"
  update-action="uninstall"
  primary-content=" was60_fp1_linux.pak"
  update-type="replace"
  log-name=
    "/opt/WebSphere/AppServer/logs/update/was60_fp1_linux.install/updatelog.txt"
  backup-name="not applicable"
  start-time-stamp="2004-12-23 15:15:14-0500"
  result="failure">
  <update-event
    event-type="component"
    parent-id="was60_fp1_linux"
    id="activity"
    update-action="uninstall"
    update-type="replace"
    start-time-stamp="2004-12-23 15:15:14-0500"
    result="failure">
  </update-event>
  <update-event
    event-type="component"
    parent-id="was60_fp1_linux"

```

```
id=" activity.impl"
update-action="uninstall"
update-type="replace"
start-time-stamp="2004-12-23 15:15:14-0500"
result="failure">
</update-event>
</update-event>
</event-history>
```

---

## Summary of Version 6 changes for the historyInfo command

Changes are in three areas: command syntax, report information, and the event.history file.

### Changes to command syntax

Version 6 replaces the term *updateID* with *maintenancePackageID* to describe a specific maintenance package. This matches the terminology used in the Version 6 update installer application.

### Changes to report information

The following changes are in effect:

- Version 6 replaces the term *PTF ID* with *Maintenance package ID*. This change is consistent with the terminology used in the Update Installer for WebSphere Software.
- Version 6 replaces the term *Primary content* with *Package file name*.
- Version 6 replaces the terms *Start Time* and *End Time* with a *Timestamp* for the completion of the event.
- Version 6 removes the terms *IsExternal*, and *IsCustom*.
- Version 6 removes the *Result Message* section of the report because the information is reported in the *Result and Action* section.
- Version 6 sets the value of *Backup file name* to "not applicable" if the update action is *uninstall*. No backup file is generated when a maintenance package is uninstalled.
- Version 6 adds a new result state, partial success. The Version 6 Update Installer for WebSphere Software can generate a partial success.

### Changes to the event.history file

The following changes are in effect:

- Version 6 replaces the terms *Start Time* and *End Time* with a *Timestamp* for the completion of the event.
- Version 6 removes the terms *IsRequired*, *IsExternal*, *IsCustom*, *root-property-file*, *root-property-name*, and *root-property-value*, which are not supported by the Update Installer for WebSphere Software.
- Version 6 removes the *Result Message* section of the report because the information is reported in the *Result and Action* section.
- Version 6 sets the value of *update-event* to "replace".
- Version 6 sets the value of *Backup file name* to "not applicable" if the update action is *uninstall*. No backup file is generated when a maintenance package is uninstalled.
- Version 6 adds a new result state, partial success. The Version 6 Update Installer for WebSphere Software can generate a partial success.



---

## Chapter 7. genVersionReport command

The **genVersionReport** command uses the **versionInfo** command to generate the `versionReport.html` report file in the current working directory, which is usually the `bin` directory. The report includes a list of changed components and installed or uninstalled maintenance packages. The `genVersionReport` script invokes the `versionInfo` script specifying the correct parameters to place the information generated into an HTML file in the current working directory.

### Product version information

The `versionInfo` tool displays important data about the product and its installed components, such as the build version and build date. History information for installation and removal of maintenance packages also displays in the report. This tool is particularly useful when working with support personnel to determine the cause of any problem.

### Product version reports

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

- `versionInfo` script  
Use the `versionInfo` command to specify your own report parameters when creating a customized version report.
- `genVersionReport` script  
Use the `genVersionReport` command to generate the `versionReport.html` report file in the current working directory, which is usually the `bin` directory. The report includes the list of components, fixes, fix packs, and refresh packs.

---

## Location of the command file

The `genVersionReport` command is a script.

The command file is named `genVersionReport` in the `bin` directory of the `app_server_root` directory.

---

## Syntax for the genVersionReport command

The command syntax is:

```
 Linux  
genVersionReport.sh
```

```
 Windows  
genVersionReport.bat
```

Issue the command from the `bin` directory of the `app_server_root` directory.

---

## Report description

The `versionInfo` command reports the following information:

## Installation information

Displays the following general information about the current installation:

- Report date and time - The date and time that the report was generated. The timestamp is formatted according to the current locale.
- Product directory - The file path to the installation root directory defined by the WAS\_HOME environment variable.
- Version directory - The file path of the version directory of the current IBM WebSphere Application Server - ND installation.
- DTD directory - The file path of the DTD directory of the current IBM WebSphere Application Server - ND installation.
- Log directory - The file path of the log directory of the current IBM WebSphere Application Server - ND installation. The maintenance package log files are in the directory.
- Backup directory - The file path of the backup directory of the current IBM WebSphere Application Server - ND installation. The backup files generated during the installation of maintenance packages are in this directory.
- TMP directory - The file path of the system temporary directory.

## Product list information

Displays a list of installed WebSphere products:

- Product ID - The product ID of the installed product.
- Status - The status of the product, either installed or uninstalled.

## Installed product information

This information and the other information topic descriptions are hierarchal for each installed product, component, component update, installed maintenance package, included APARs, and component updates.

This section of the report displays the following information:

- Name - The name of the installed product.
- Version - The current version of the product. Installing or uninstalling fix packs or refresh packs modifies this version.
- ID - The product ID of the product installed, such as BASE, BASETRIAL, ND, EXPRESS, EXPRESSTRIAL, embeddedEXPRESS, IHS, XD, PLG, or CLIENT.
- Build level - The build level of the installed product.
- Build date - The build date of the installed product.

### Installed component information

Displays the following component-level information of the installed component from the .component file in the *app\_server\_root*/properties/version directory:

- Component name - The name of the installed component
- Spec version - The spec version of the current component
- Build level - The build level of the current component
- Build date - The build date of the current component

**Installed component update information:** Displays the general maintenance package information:

- Component name - the name of the installed component
- Update type - All interim fixes, fix packs and refresh packs are maintenance packages. The update type field is always set to *maintenance package*.

- Maintenance package ID - The ID of the maintenance package that is responsible for the current maintenance level of the product.
- Update effect - The updated action taken on a particular component. The default effect is *replace*.
- Log file name - The file path of the log file that records the maintenance actions for the current maintenance level.
- Backup file name - The file path of the backup file generated during the installation of the maintenance package. Not applicable if the component is restored as a result of an uninstall of a maintenance package.
- Timestamp - The time at which the component is updated. The timestamp is stated in GMT offset values.

*Installed maintenance package information:* Displays the general maintenance package information:

- Maintenance Package ID - the maintenance package ID
- Description - the description of the maintenance package
- Build Date - the build date of this maintenance package

*Included APARs information:* Displays the list of APARs fixed by this maintenance package.

*Component updates information:* Displays the following information about each component that is updated by the installed maintenance package:

- Component name - The name of the installed component.
- Updated effect - The update action taken on the component by the maintenance package. The default action is *replace*.
- Timestamp - The time at which the component update is installed (GMT offset).

---

## Summary of Version 6 changes for the VersionInfo command

Changes are in two areas: command syntax and report information.

### Changes to command syntax

The following changes are in effect:

- Version 6 replaces the terms *efixes* and *ptfs* with *maintenancePackages* to describe a specific maintenance package. This matches the terminology used in the Version 6 update installer application.
- Version 6 replaces the terms *efixesDetail* and *ptfDetail* with *maintenancePackageDetail* to describe the detailed information that relates to a specific maintenance package.

### Changes to report information for Version 6.x

The following changes are in effect:

- Version 6 replaces the term *technology list* with *product list*. The technology list displays the list of installed products in the current WAS\_HOME directory.
- Version 6 uses an update type value of *maintenance package*.
- Version 6 replaces the term *PTF ID* with *Maintenance Package ID*. This change is consistent with the terminology used in the Update Installer for WebSphere Software.

- Version 6 removes *IsRequired*, *IsOptional*, *IsExternal*, and *IsCustom* from the *Installed component update* section of the report. The older terminology is not supported per component by the Version 6 Update Installer for WebSphere Software.
- Version 6 replaces the term *Install Date* with *Timestamp* for consistency with the History report.
- Version 6 replaces the *Installed PTF* section of the report with the *Installed Maintenance Package* section.
- Version 6 removes the term *Exposition* because it is similar to the term *Description*.
- Version 6 removes the *Build Version* field from the *Installed Maintenance Package* section.
- Version 6 removes the *Supported Platforms* section of the report.
- Version 6 replaces the *Included Fixes* section of the report with the with by *Included APARs* section, which lists the APARs.
- Version 6 removes the *Custom Properties* section of the report.

---

## Chapter 8. Product version information

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

Run the `historyInfo` command to create a report about installed maintenance packages. The `historyInfo` command creates a report on the console and also creates tracking files in the `app_server_root/properties/version/history` directory.

Time-stamped, detailed logs record each update process in the `properties/version/log` directory of the `app_server_root`.

This topic describes the XML data files that store product information for Version 6 WebSphere Application Server products. By default, the document type declarations (DTDs) for these files are in the `properties/version/dtd` folder of the `app_server_root`, or the server root directory. See the Chapter 8, "Product version information" section for more information.

This topic includes the following sections:

- "Product information files"
- "Reports" on page 48
- "Logs and component backups" on page 49
- "Directory locations" on page 50
- "Operational description" on page 50
- "Data dictionary" on page 51

---

### Product information files

#### XML files in the `properties/version` directory that store version information: `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="6.1"/>
```

The following XML files in the `properties/version` directory represent installed items and installation events such as product edition, version, component, and build information.

#### `WAS.product`

One file whose existence indicates the particular WebSphere Application Server product that is installed. The type of product installed is indicated by the `<id>` tag. Data in the file indicates the version, build date, and build level.

For example, `<id>ND</id>.product` 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 - ND">
```

```

    <id>ND</id>
    <version>6.1.0</version>
    <build-info date="02/03/06" level="s0461.18"/>
</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="05/31/06" build-version="a0522.07"
  name="activity" spec-version="6.1.0.0"/>

```

#### *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.

#### *ptf-id.ptf*

Any number of maintenance package files that each indicate the presence of an installed refresh pack, fix pack, or interim fix.

**XML files in the `properties/version/history` directory that store version history information files:** The following XML files in the `properties/version/history` directory describe refresh packs, fix packs, and interim fixes that are currently installed. These XML files are related to installation items by the primary ID information, which is shown in the following examples as italicized text.

#### *xxxmaintenance\_package\_identifiersxxx.ptfDriver*

A driver file has useful information about the entire contents of an interim fix or fix pack.

#### *xxxmaintenance\_package\_identifiersxxx.ptfApplied*

The applied file has relevant information about the interim fixes or fix packs that are currently applied.

#### **event.history**

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

---

## Reports

WebSphere Application Server provides the ability to generate *Version* reports and History reports from the data in the files. The following report-generation scripts are available in the `app_server_root` bin directory.

### **Product version reports**

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

- `versionInfo` script  
Lets you use parameters to create a version report on Linux and UNIX platforms, or on Windows platforms.
- `genVersionReport` script

Generates the versionReport.html report file in the bin directory on Linux and UNIX platforms, or on Windows platforms. The report includes the list of components and installed and uninstalled maintenance packages.

### Product history reports

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

- historyInfo script  
Lets you use parameters to create a history report on Linux and UNIX platforms, or on Windows platforms.
- genHistoryReport script  
Generates the historyReport.html report file in the bin directory on Linux and UNIX platforms, or on Windows platforms. The report includes the list of components and a history of installed and uninstalled maintenance packages.

---

## Logs and component backups

WebSphere Application Server products use two other directories when performing update operations, for logging and backups:

*app\_server\_root* /logs/update

The logs directory for product updates.

The location of log files that describe events that occur during the use of the update installer program.

*app\_server\_root*properties/version/backup

Product updates backup directory

WebSphere Application Server products back up components before applying interim 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: 20050324\_211832 is 24-Mar-2004, 9:18:32 pm, GMT. All time stamps are in GMT.

**ID** Interim fix ID or fix pack ID

For example: apar6789c is an interim fix ID; PTF\_1 is a fix pack ID.

#### Operation

install | uninstall

#### Interim fix log file names

*timeStamp\_fixID\_operation.log*

For example, the Update installer program creates these logs:

*app\_server\_root*logs/update/20050324\_211832\_apar6789c\_install.log and  
*app\_server\_root*/logs/update/ 20050324\_211912\_apar6789c\_uninstall.log

#### Interim fix component log file names

*timeStamp\_fixId\_componentName\_operation.log*

For example, the update installer program creates these logs:

*app\_server\_root*/logs/update/20050324\_211832\_apar6789c\_ras\_install.log  
and *app\_server\_root*/logs/update/  
20050324\_211912\_apar6789c\_ras\_uninstall.log

### Fix pack log file names

*timeStamp\_ptfId\_operation.log*

For example, the update installer program creates these logs:

*app\_server\_root/logs/update/20050924\_211832\_was60\_fp1\_install.log* and  
*app\_server\_root/logs/update/20050924\_211912\_was60\_fp1\_uninstall.log*

### Fix pack component log file names

*timeStamp\_ptfId\_componentName\_operation.log*

For example, prior to Fix Pack 2: *properties/version/log/*

*20050324\_211832\_was50\_fp1\_ras\_install.log* and *properties/version/log/*  
*20030325\_211912\_was50\_fp1\_ras\_uninstall.log*The update installer program  
creates these logs: *app\_server\_root/logs/update/*  
*20050324\_211832\_was60\_fp1\_ras\_install.log* and *app\_server\_root/logs/*  
*update/20030325\_211912\_was60\_fp1\_ras\_uninstall.log*

### Backup JAR file names

*timeStamp\_ptfId\_componentName\_undo.jar* or

*timeStamp\_fixId\_componentName\_undo.jar*

For example: *20020924\_211832\_apar6789c\_ras\_undo.jar*Do not delete a  
backup 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.

---

## Directory locations

Product information files are located relative to the WebSphere Application Server  
product *app\_server\_root*, or the server root directory.

Default file paths are:

#### Version directory

*app\_server\_root/properties/version*

#### History directory

*app\_server\_root/properties/version/history*

#### Updates log directory

The update installer program stores log files in the *app\_server\_root/logs/*  
*update* directory.

#### Updates backup directory

*app\_server\_root/properties/version/backup*

#### DTD directory

*app\_server\_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 removes an interim fix file from the  
version directory when it uninstalls the corresponding fix.
- A WebSphere Application Server product adds a file with an extension of `.ptf`  
to the version directory to indicate that a refresh pack, a fix pack, or an interim fix  
is currently installed.

- A WebSphere Application Server product removes a file with an extension of .ptf from the version directory when it uninstalls the corresponding refresh pack, a fix pack, or an interim fix.
- A WebSphere Application Server product adds a driver file with an extension of .ptfDriver to the version/history directory when you run the historyInfo command. A fix pack driver file contains defining information for a fix pack.
- A WebSphere Application Server product adds a fix pack applied file with an extension of .ptfApplied to the version/history directory when you run the historyInfo command. 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 a maintenance package.
- A WebSphere Application Server product writes a line about a parent event for each refresh pack, a fix pack, or interim fix 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 interim fix, fix pack, or refresh pack parent event.
- A WebSphere Application Server product stores one log file in the logs/update directory as it installs or uninstalls one interim fix, fix pack, or refresh pack.
- A WebSphere Application Server product stores one log file in the logs/update directory as it installs or uninstalls an interim fix, fix pack, or refresh 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 denotes 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*/WAS.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 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.

Type Family: update

File Types: ptf  
ptf-applied

File Type: ptf

Persistence: *versionDir/id.ptf*

Elements:	id	string	required
	short-description	string	required
	build-version	string	required
	build-date	date	required
	component-name	complex	min=1, max=unbounded

Type Detail:

A ptf file denotes the presence of some portion of a specific refresh pack, fix pack, or interim fix.

The id of the refresh pack, fix pack, or interim fix is embedded in the fix pack file name.

A ptf file contains a listing of component updates.

When installing a refresh pack, fix pack, or interim fix, you can omit certain potential component updates, but only when the corresponding component is not installed.

Examine a separate application file to determine the components that a particular refresh pack, fix pack, or interim fix updates.

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

Element Detail:

ptf.id	The ID of the fix pack.
ptf.short-description	A short 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-name	A list of components.

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 refresh pack, fix pack, or interim fix as specified by the ID.

Element Detail:

ptf-applied.ptf-id	The ID of the refresh pack, fix pack, or interim fix for which applies 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]
	log-name	anyURL	required
	backup-name	anyURL	required
	time-stamp	date	required

Type Detail:

An applied instance is present to indicate the application of an update for a particular interim fix, fix pack, or refresh pack to a particular component.

(The particular interim fix, fix pack, or refresh pack is specified by the applied 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.log-name	The name of the log file that 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).

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 refresh pack, fix pack, and interim fix 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]
	primary-content	anyURI	required	
	update-action	enum	required	[enumEventAction]
	log-name	anyURI	required	
	backup-name	anyURI	required	
	start-time-stamp	dateTime	required	
	result	string	required	
	update-event	complex	optional	

Type Detail:

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

Element Detail:

update-event.event-type The type of this event, either a refresh pack, fix pack, or an interim fix type event, or a component type event.

update-event.parent-id This element is present only for component events. The ID of the parent interim fix, fix pack, or refresh pack of this event.

update-event.id The ID of the interim fix, fix pack, refresh pack, or component that was updated, interpreted according to the type of the event.

update-event.update-type The type of update for an update event.

update-event.update-action The type of action for this event.

update-event.log-name The name of the log file that was generated for this event.

update-event.backup-name	The name of the backup file that 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.result	The result of the update.
update-event.update-event	A collection of child events. This collection is used for interim fix and fix pack type events. This collection is empty for component type events.

Type Detail:

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

Value Detail:

EventType.ptf Specifies that an event is for a refresh pack, fix pack, or interim fix update.

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

Enum Type: update-action

Values: 0 Install  
1 Uninstall

Type Detail:

An event action instance specified the operation performed by an update, which can be an install or uninstall operation.

Value Detail:

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

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

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.



---

## Appendix A. Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
500 Columbus Avenue  
Thornwood, New York 10594 USA



---

## **Appendix B. Trademarks and service marks**

For trademark attribution, visit the IBM Terms of Use Web site (<http://www.ibm.com/legal/us/>).