

# **WebSphere Application Server 7.0 AIX Migration**

# Contents

<b>Introduction</b> .....	<b>4</b>
<b>Planning the migration to WAS 7.0</b> .....	<b>7</b>
<b>About migrating applications</b> .....	<b>8</b>
<b>Creating a new WAS 7.0 test environment</b> .....	<b>9</b>
Installing WAS 7.0 .....	10
Updating WAS 7.0 to the latest maintenance level.....	10
Installing the IBM Update Installer for WebSphere Software.....	11
Applying a fix pack to WAS 7.0 .....	11
Applying a fix pack to the WAS 7.0 SDK .....	12
Creating a deployment manager profile.....	13
Creating a profile for a managed node.....	13
Creating an application server cluster .....	14
<b>Migrating a WAS 6.1 test environment to WAS 7.0</b> .....	<b>14</b>
Migrating WAS 6.1 to 7.0 .....	15
Installing WAS 7.0 .....	15
Updating WAS 7.0 to the latest maintenance level.....	16
Migrating the WAS configuration.....	18
<b>Migrating a WAS 6.1 production environment to WAS 7.0</b> .....	<b>26</b>
Migrating WAS 6.1 to 7.0 .....	26
Installing WAS 7.0 .....	26
Updating WAS 7.0 to the latest maintenance level.....	27
Migrating the WAS configuration.....	30
Migrating IBM HTTP Server (IHS) 6.1 to 7.0.....	38
Installing IBM HTTP Server 7.0.....	38
Installing the IBM HTTP Server for WebSphere Application Server plug-in 7.0 .....	39
Configuring the web server plug-in.....	40
<b>Considerations for integration with other products</b> .....	<b>42</b>
DB2 considerations .....	42
WebSphere MQ considerations.....	43
ITCAM considerations .....	44
Updating ITCAM for WebSphere.....	44
Additional notes on updating ITCAM for WebSphere.....	48
<b>Rolling back to WAS 6.1</b> .....	<b>50</b>
<b>Troubleshooting</b> .....	<b>51</b>
Problems starting the migrated environment .....	51
Problems connecting to WebSphere MQ .....	52
Problems after changing WMQ Native Library Path .....	52
Problems connecting to WMQ when using 64 bit WAS .....	53
<b>References</b> .....	<b>53</b>
<b>Appendices</b> .....	<b>54</b>
Installing WAS 6.1 .....	54

Updating WAS 6.1 to the latest maintenance level.....	55
Installing the IBM Update Installer for WebSphere Software.....	55
Applying a fix pack to WAS 6.1.....	56
Applying a fix pack to the WAS 6.1 SDK .....	57
Notices .....	58

# Introduction

---

## About this project

This document covers the migration of a WebSphere Application Server completed as part of the Mixed Platform Stack project. This project is based on a customer scenario used to perform integration testing of a variety of IBM software products. The scenario is based on a fictitious insurance company (LGI) that provides insurance quotes and policies to consumers through web applications.

Further information about the Mixed Platform Stack project can be found at "[The Mixed Platform Stack Project: Deploying a secure SOA solution into z/OS and mixed z/OS and AIX environments v2.0](#)."

In the scenario, the LGI company is migrating to WebSphere Application Server 7.0 to gain the following business value:

- A method to ensure that the customer applications are running in a recent, supported environment.
- The ability to use significant new and updated functionality available in the new release.

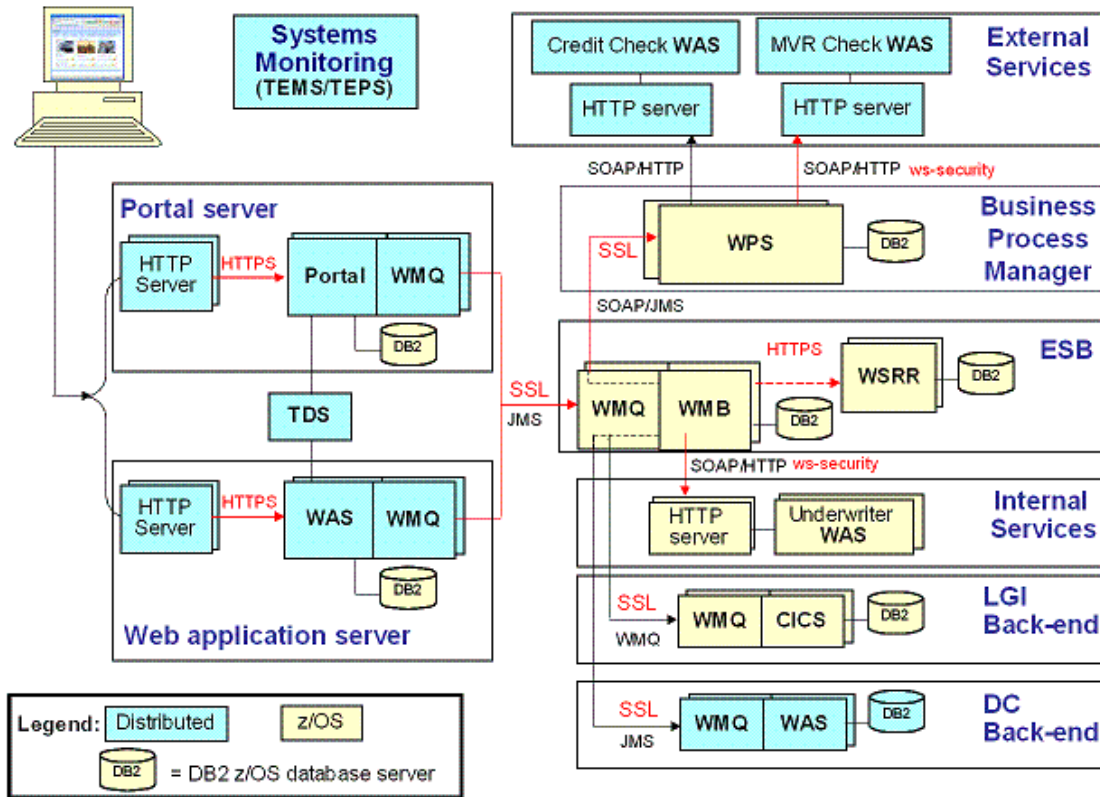
WebSphere Application Server is used on distributed platforms in the scenario to provide the presentation layer used by consumers. This is the primary business route for customer interactions with the LGI company. The company is therefore highly dependent on the WebSphere Application Server environment and has imposed the following restrictions on the migration:

- Maintain availability of service to consumers.
- Minimize disruption of in progress transactions to consumers.
- Minimize updates required to application code for compatibility with new release to reduce risk of regressions in application behavior due to code changes.
- Minimize changes required to supporting infrastructure.

As stated previously, the WebSphere Application Server environment serves as the presentation layer in the scenario. The presentation layer communicates with other components that provide vital business applications. Technologies such as EJBs and enterprise messaging are used to communicate with these other components. Communications with these business applications must remain operational after the migration. The presentation layer communicates with the following components:

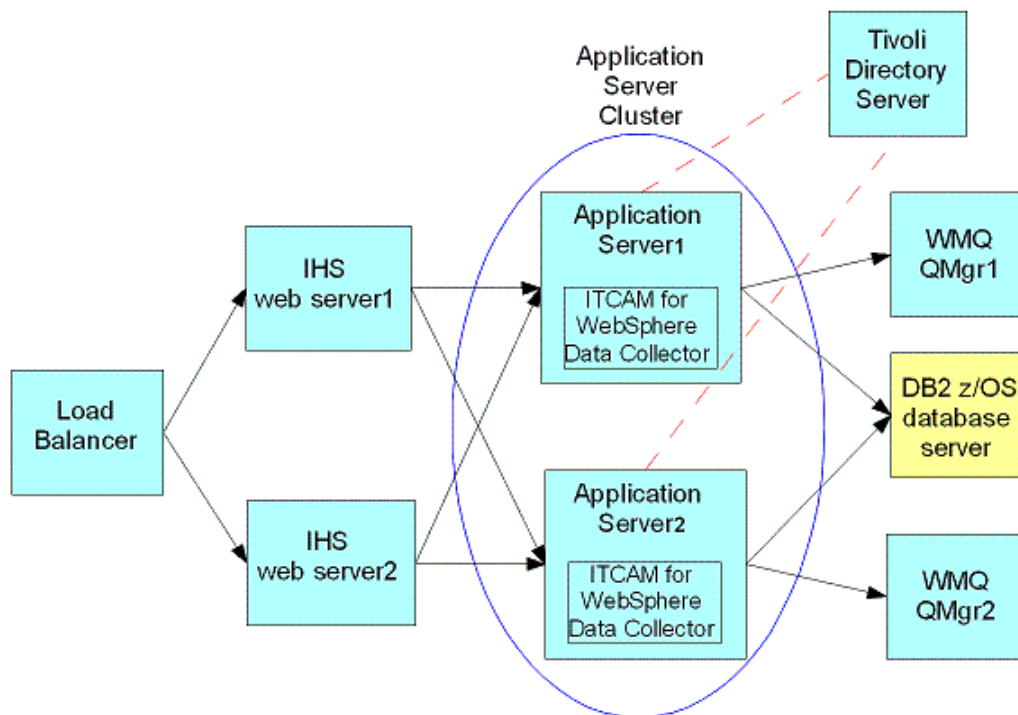
- DB2 databases running on z/OS
- WebSphere MQ queue managers running locally through bindings mode connections
- IBM Tivoli Composite Application Manager monitoring agents
- An LDAP user directory implemented using IBM Tivoli Directory Server

The following diagram describes the LGI company environment used by the Mixed Platform Stack project:



**Figure 1: LGI company environment**

The following diagram describes the WebSphere Application Server configuration used in this project and shows the integration with other products in the solution:



**Figure 2: LGI company's WAS configuration**

Testing was performed in the following environment:

- Operating System:
  - AIX 5300-08-01-819
- Software:
  - WAS ND 6.1.0.21 to WAS ND 7.0.0.1
  - WMQ 6.0.2.4 to WMQ 6.0.2.5
  - Remote database server: DB2 9.1 for z/OS
  - ITCAM for WebSphere 6.1.0.3 to ITCAM for WebSphere 6.1.0.4
- Configuration:
  - WAS ND Cluster with one deployment manager, two managed nodes each with one application server
- Applications:
  - Enterprise Applications with Web modules (HTML, JSPs and servlets) and EJB modules (Session beans making JMS API calls to WebSphere MQ and JDBC API calls to DB2 and Entity beans accessing DB2)

### About this team

This project is a collaborative project involving teams from IBM Software Group (SWG) and IBM Systems and Technology Group (STG).

The SWG Federated Integration Test (FIT) team is a worldwide team whose mission is to test the integration of IBM products and to provide feedback to IBM product teams, customer-facing teams, and customers. The FIT team uses realistic, customer-like scenarios to perform integration testing on stacks of products.

The STG z/OS Integration Test team, also known as zPET, is responsible for the final phase of IBM System z product testing. The team's mission is to validate z/OS functionality in an environment that closely simulates realistic production workloads and focuses on typical customer tasks.

### About the migration process

This document provides information about migration from WAS 6.1 to 7.0 on distributed platforms, including detailed information on the steps required. The document covers:

- Migration of WAS ND Cluster running as either root or a non-root user
- Migration of WAS configuration integrating with a remote DB2 database server on z/OS
- Migration of WAS configuration integrating with WebSphere MQ (WMQ) in BINDINGs mode
- Migrating while maintaining a service

**Note:** The information documented here is one method of performing this migration and provides information on points at which you may wish to select other options to gain further granular control. Other migration options are available.

This project used the following approach to perform the WAS migration:

1. Planning the migration
2. Migrating applications for compatibility with WAS 7.0
3. Creating a WAS 7.0 test environment
4. Migrating a test environment from WAS 6.1 to 7.0
5. Migrating a production environment from WAS 6.1 to 7.0

## Planning the migration to WAS 7.0

---

This section describes several key aspects in planning a migration.

The planning phase is vital to the success of a migration; attention to detail during the planning phase will enable the team to avoid many potential problems. This section focuses on the technical considerations for a migration, but planners are also advised to consider the timely availability of key personnel who are skilled with WAS 7.0.

As part of your migration you should take an inventory of your current WebSphere Application Server environment. Collect the following information:

- The environments where WebSphere Application Server is currently deployed (for example, Development, Test, Production)
- The versions of WebSphere Application Server currently deployed
- The specifications of the machines running your WebSphere Application Server environments
- The applications currently deployed in your environments
- The other software products that WebSphere Application Server integrates with in your environment
- The business criteria that may affect the migration (for example, frequency and length of outage periods, service level agreements)

The information you collect will be a vital part of the successful completion of a migration project. Some of the potential uses for this information include:

- The list of environments will be useful in determining the number of migration activities that will need to take place. Assessing the size and complexity of each of these environments will allow you to gauge the approximate time that will be needed to complete the migration of those environments. You should also use this information to determine the sequence in which you will migrate the environments and to determine whether any additional environments will be required for testing purposes.
- The versions of WebSphere Application Server currently deployed will determine the migration path that you will follow. You should verify that there is a supported migration path from your current

version of the product to version 7.0. For more information see the [Migrating product configurations](#) section of the WebSphere Application Server infocenter.

- The specifications of the machines running in your current environment will be used to determine compatibility with the new version of the product. If these machines are not compatible with the new release, it may be necessary to build a new environment to host version 7.0 of WAS.
- The list of applications to be migrated will also provide a good opportunity for you to verify that all of your currently deployed applications are still required. You might be able to remove some redundant applications prior to the migration.
- The list of other software products that WebSphere Application Server integrates with is needed to help determine any dependencies between those additional products.
- The business criteria will help determine the schedule used for your migration and the level of flexibility you will have for system outages during the migration.

You should also consider the effort that will be required to test your applications and environment on the new version of WAS; it would be risky to assume that the applications will work as they did previously. You should include performance testing during your testing as the performance characteristics of WAS may change between releases. You should also consider the impact of other changes such as moving from a previous level that supported only 32-bit systems to a 64-bit version of the current release as this may also change the performance characteristics.

Prior to commencing any migration activity, you should ensure that working backups are in place and just as importantly ensure that you are able to successfully restore from these backups. This will ensure that you are able to restore a working system should you encounter major problems.

Another vital consideration is the development of a contingency plan should any serious problems occur during the migration. You should develop a rollback plan prior to attempting the migration and test the rollback procedure on a non-critical environment. Should the migration of your production environment fail, you will need to rollback to the previous working level of WAS. Testing this aspect of the migration will allow you to gain experience with the rollback procedure and ensure that you can execute it quickly and correctly in the unlikely event that it is required.

## About migrating applications

---

This section gives a brief overview of the impact of the migration on applications and considerations to help migrate them.

The focus of this whitepaper is the migration of the runtime environment and not the application development environment. This section does not cover the application migration aspects in detail but does provide an overview of many of the key issues relating to application migration.

A major consideration during a migration is the impact to applications that are currently deployed on the existing version of WAS. These applications may make use of features that are not present in more recent releases of WAS.

There is great focus on ensuring application compatibility between different versions of WAS, and it is possible that your application may continue to run without modification.

### **Determining whether existing applications are supported in WAS 7.0**

Before attempting the migration, you should review the list of deprecated, stabilized and removed features in WAS 7.0. You should examine your applications to determine whether the applications use functionality that has been removed from the target version of WAS. In a case where an application is dependent on a feature that has been removed from WAS, the application will have to be changed to work with an alternative feature.

Features that have been listed as deprecated will continue to function, however, you should plan to change your application in the future such that it will continue to operate without the deprecated feature prior to its potential removal.

The list of features that may no longer be available is highly dependent on the source version of WAS being migrated from and the target version of WAS being migrated to. In general, the more recent your current version of WAS, the better the chance that your applications will continue to run without modification.

The level of the J2EE specification supported by WAS can change between releases. Typically, an application built to an older level of the specification will still work with a newer version of WAS without changing the specification level used by the application. It is easy to change the target runtime and hence J2EE specification level in the development tooling, should this be necessary.

### **Recommendations for changing applications**

Any changes that must be made to applications should be kept to the minimum possible and ideally the minimum required to allow correct operation with the new release. This reduces the possibility that new errors are introduced during the migration process. Minimizing the number of changes also facilitates troubleshooting, enabling the tester to pinpoint the cause of any errors.

New functionality is frequently the driver for migrating to a newer release of WAS. We recommend that changes to applications to take advantage of any new functionality are postponed until after the migration process has been completed. In cases where the use of new functionality is desirable, it may be necessary to upgrade your development tools to a newer level. You should review the development features provided by your development tools to determine whether an upgrade is required.

When you attempt to move projects from the old version of your development environment to the new version, you should always work with copies of resources such as workspaces. This will enable you keep a known, working copy of your application assets should anything go wrong when you move to the newer version.

### **Specific consideration for this project**

In the scenario covered in this document, the currently deployed version of WebSphere Application Server is 6.1 and the target version is 7.0. The applications deployed on version 6.1 are applications that have been developed over time and for multiple versions of WAS. The current application environment is the result of several previous migrations. The applications tested in this environment have been continually updated to ensure that they do not make use of deprecated or removed WAS features. As a consequence, the applications are able to run without modification on version 7.0 of WAS. Once the applications had been tested and verified without modification, the applications were updated to use the latest J2EE specifications offered by version 7.0. This is in preparation for the addition of code changes to take advantage of the new functionality provided by this version of WAS.

## **Creating a new WAS 7.0 test environment**

---

This topic describes the creation of a new WAS 7.0 test environment.

There are several benefits to creating a new test environment as part of your migration. These include:

- Allows you to gain experience with the new version of the product
- Gives you an environment to test that your applications work correctly
- Helps you to determine whether any problems are introduced by the new version or the migration process

The high level steps to create a WAS 7.0 test environment are as follows:

1. Install WAS 7.0.
2. Update WAS 7.0 to the latest maintenance level.
3. Create a deployment manager profile.
4. Create managed profiles for the nodes.
5. Create an application server cluster.

Once you have created a new WAS 7.0 test environment, you can use it to test and verify as necessary.

## Installing WAS 7.0

---

This task describes the steps necessary to install WebSphere Application Server Network Deployment 7.0.

Use the following procedure to install the WebSphere Application Server Network Deployment product.

This procedure installs the product ready for use in a new environment or allows migration from a previous version. Run all commands as root user.

1. Obtain the install media appropriate for your platform. For example: Download part number **C1G2QML** for 64-bit AIX.
2. Enter the following command to create a temporary directory to hold the install media:

```
mkdir /tmp/was7
```

3. Copy the software image to the temporary directory.

4. Enter the following commands to extract the install media into the temporary directory:

```
gzip -d C1G2QML.tar.gz
```

```
tar -xvf C1G2QML.tar
```

5. Use a text editor to create a response file called `responsefile.txt` in `/tmp/was7/WAS` with the following content:

```
-OPT silentInstallLicenseAcceptance="true";
-OPT installType="installNew";
-OPT profileType="none";
-OPT feature="noFeature";
-OPT installLocation="/usr/IBM/WebSphere70/AppServer"
```

6. Enter the following commands to start the installer in silent mode using the response file created in the previous step:

```
cd /tmp/was7/WAS
```

```
./install -options "/tmp/was7/WAS/responsefile.txt" -silent
```

**Note:** If serious problems are encountered that prevent the product from installing, the errors are logged in the `/waslogs` directory. Review the files in this directory, correct any problems (for example insufficient space on target file systems), and retry the installation.

- a) Verify that the installation succeeded using the `installver.sh` command. This command verifies that the product files were installed as expected and reports on any variations.

```
/usr/IBM/WebSphere70/AppServer/bin/installver.sh
```

7. If the installation files are no longer needed, they can be deleted by running the following command:

```
rm -r /tmp/was
```

The product files are installed into the `/usr/IBM/WebSphere70/AppServer` directory on the target machine.

## Updating WAS 7.0 to the latest maintenance level

---

This topic describes the high level steps necessary to update WebSphere Application Server Network Deployment 7.0 to the latest maintenance level.

The high level steps required to update the WAS ND 7.0 to the latest maintenance level are:

1. Install the latest version of the IBM Update Installer for WebSphere Software.
2. Apply the latest WAS fix pack.
3. Apply the latest WAS SDK fix pack.

Details for each of these steps are provided in the sections that follow.

## Installing the IBM Update Installer for WebSphere Software

This task describes the steps necessary to install the IBM Update Installer for WebSphere Software. This software is used to install maintenance and updates to WebSphere Application Server.

Use the following procedure to install the IBM Update Installer for WebSphere Software. Run all commands as root user.

1. Obtain the install file appropriate for your platform from <http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg24020212>.

a) For example: download 7.0.0.1-WS-UPDI-AixPPC64.tar.gz for 64 bit AIX.

2. Enter the following command to create a temporary directory to hold the install media:

```
mkdir /tmp/updi
```

3. Copy the software image to the temporary directory.

4. Enter the following commands to extract the install media to the temporary directory:

```
gzip -d 7.0.0.1-WS-UPDI-AixPPC64.tar.gz
```

```
tar -xvf 7.0.0.1-WS-UPDI-AixPPC64.tar
```

5. Use a text editor to create a response file called `responsefile.txt` in `/tmp/updi/UpdateInstaller` with the following content:

```
-OPT silentInstallLicenseAcceptance="true"  
-OPT installLocation="/usr/IBM/WebSphere70/UpdateInstaller"
```

6. Enter the following commands to start the installer in silent mode using the response file created in the previous step:

```
cd /tmp/updi/UpdateInstaller
```

```
./install -options "/tmp/updi/UpdateInstaller/responsefile.txt" -silent
```

**Note:** If serious problems are encountered that prevent the product from installing, the errors are logged in the `/updilogs` directory. Review the files in this directory, correct any problems (for example insufficient space on target file systems), and retry the installation.

a) Verify that the installation succeeded using the `installver.sh` command. This command verifies that the product files were installed as expected and reports on any variations.

```
/usr/IBM/WebSphere70/UpdateInstaller/bin/installver.sh
```

7. If the installation files are no longer needed, they can be deleted by running the following command:

```
rm -r /tmp/updi
```

The software is installed into the `/usr/IBM/WebSphere70/UpdateInstaller` directory on the target machine.

## Applying a fix pack to WAS 7.0

This task describes the steps necessary to use the IBM Update Installer for WebSphere Software to update WebSphere Application Server Network Deployment 7.0 with the latest recommended fix pack.

The list of recommended fixes for all versions of WebSphere can be found at the following location:

<http://www-01.ibm.com/support/docview.wss?uid=swg27004980>

At the time of writing, the most recent available fix pack for WebSphere Application Server 7.0 is Fix Pack

1. The instructions below apply to that fix pack.

Use the following procedure to apply a fix pack to WAS 7.0. Run all commands as root user.

1. Obtain the appropriate fix pack file for your platform.

a) For example, download 7.0.0-WS-WAS-AixPPC64-FP0000001.pak for 64 bit AIX.

2. Copy the fix pack file to the maintenance directory under the Update Installer directory which is `/usr/IBM/WebSphere70/UpdateInstaller/maintenance`

- Use a text editor to create a response file called `install7.0.0.1.txt` in the directory `/usr/IBM/WebSphere70/UpdateInstaller/responsefiles` with the following content:

```
-W product.location="/usr/IBM/WebSphere70/AppServer"
-W maintenance.package="/usr/IBM/WebSphere70/UpdateInstaller/
maintenance/7.0.0-WS-WAS-AixPPC64-FP0000001.pak"
-W update.type="install"
```

- Enter the following commands to start the installer in silent mode using the response file created in the previous step:

```
cd /usr/IBM/WebSphere70/UpdateInstaller
```

```
./update.sh -options "/usr/IBM/WebSphere70/UpdateInstaller/responsefiles/
install7.0.0.1.txt" -silent
```

- Enter the following command to run the VersionInfo tool:

```
/usr/IBM/WebSphere70/AppServer/bin/versionInfo.sh
```

- Ensure that the update was applied by checking that the Version field correctly reflects the fix pack you applied, in this case 7.0.0.1. The output should be similar to the following:

```
WVER0010I: Copyright (c) IBM Corporation 2002, 2005, 2008; All rights
reserved.
WVER0012I: VersionInfo reporter version 1.15.1.26, dated 8/9/08
```

```
-----
IBM WebSphere Application Server Product Installation Status Report
-----
```

```
Report at date and time 06 January 2009 14:45:41 GMT
```

```
Installation
-----
```

```
Product Directory /usr/IBM/WebSphere70/AppServer
Version Directory /usr/IBM/WebSphere70/AppServer/properties/version
DTD Directory /usr/IBM/WebSphere70/AppServer/properties/version/dtd
Log Directory /usr/IBM/WebSphere70/AppServer/logs
Backup Directory /usr/IBM/WebSphere70/AppServer/properties/version/nif/
backup
TMP Directory /tmp
```

```
Product List
-----
```

```
ND installed
```

```
Installed Product
-----
```

```
Name IBM WebSphere Application Server - ND
Version 7.0.0.1
ID ND
Build Level cf010845.08
Build Date 11/12/08
Architecture PPC64
-----
```

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

WAS ND 7.0 is upgraded to the latest maintenance level on the target machine.

## Applying a fix pack to the WAS 7.0 SDK

This task describes how to use the IBM Update Installer for WebSphere Software to update WebSphere Application Server 7.0 Software Development Kit (SDK) with the latest recommended fixes.

The list of recommended fixes for all versions of WebSphere can be found at the following location:

<http://www-01.ibm.com/support/docview.wss?uid=swg27004980>

At the time of writing, the most recent available Java SDK fix pack for WebSphere Application Server 7.0 is Fix Pack 1. The instructions below are applicable for that fix pack.

Use the following procedure to apply a fix pack to WAS 7.0 SDK. Run all commands as root user.

1. Obtain the appropriate fix pack file for your platform.
  - a) For example download 7.0.0-WS-WASSDK-AixPPC64-FP0000001.pak for 64 bit AIX
2. Copy the fix pack file to the maintenance directory under the Update Installer directory which is `/usr/IBM/WebSphere70/maintenance`
3. Using a text editor create a response file called `installSDK7.0.0.1.txt` in the directory `/usr/IBM/WebSphere70/UpdateInstaller/responsefiles` with the following content:

```
-W product.location="/usr/IBM/WebSphere70/AppServer"
-W maintenance.package="/usr/IBM/WebSphere70/UpdateInstaller/
maintenance/7.0.0-WS-WASSDK-AixPPC64-FP0000001.pak"
-W update.type="install"
```

4. Run the update program using the response file created in the previous step:

```
cd /usr/IBM/WebSphere70/UpdateInstaller
```

```
./update.sh -options "/usr/IBM/WebSphere70/UpdateInstaller/responsefiles/
installSDK7.0.0.1.txt" -silent
```

5. Verify that the update was applied using the version information command:

```
/usr/IBM/WebSphere70/AppServer/java/bin/java -fullversion
```

- a) The output should be similar to the following. Ensure that the Version field correctly reflects the Java SDK fix pack you applied. For example, the following message will be displayed for the 7.0.0.1 Java SDK

```
java full version "JRE 1.6.0 IBM AIX build pap6460sr3-20081106_07 (SR3)"
```

WAS 7.0 SDK is upgraded to the latest maintenance level.

## Creating a deployment manager profile

---

This task describes the steps necessary to create a deployment manager profile for a WAS 7.0 environment.

You have a new WebSphere Application Server 7.0 installation with no profiles. Use the following procedure to create a deployment manager profile. Run all commands as root user.

Enter the following commands to create a WAS 7.0 deployment manager profile:

For example: To create a deployment manager profile called `TestDmgr01` with administration security enabled

```
cd /usr/IBM/WebSphere70/AppServer/bin
```

```
./manageprofiles.sh -create -templatePath /usr/IBM/WebSphere70/AppServer/
profileTemplates/management -profileName TestDmgr01 -profilePath /usr/
IBM/WebSphere70/AppServer/profiles/TestDmgr01/ -enableAdminSecurity true
-adminUserName wasadmin -adminPassword password -cellName TestCell01 -
nodeName TestDmgr01
```

A deployment manager profile called `TestDmgr01` is created in `/usr/IBM/WebSphere70/AppServer/profiles/TestDmgr01`.

## Creating a profile for a managed node

---

This task describes the steps necessary to create a profile for a managed node in a WAS 7.0 environment.

You have a new WebSphere Application Server 7.0 installation with a deployment manager profile. Use the following procedure to create a profile for a managed node. Run all commands as root user.

Enter the following commands to create a WAS 7.0 profile for a managed node:

For example: To create a profile for a managed node called `TestNode01`

```
cd /usr/IBM/WebSphere70/AppServer/bin
```

```
./manageprofiles.sh -create -templatePath /usr/IBM/WebSphere70/AppServer/
profileTemplates/managed -profileName TestNode01 -profilePath /usr/IBM/
```

```
WebSphere70/AppServer/profiles/TestNode01 -nodeName TestNode01 -cellName
TestNode01 -federateLater true
```

**Note:** Do NOT federate the managed node yet, this will be performed as part of creating a WAS 7.0 cluster.

A managed node profile called TestNode01 is created in /usr/IBM/WebSphere70/AppServer/profiles/TestNode01.

## Creating an application server cluster

This task describes the steps necessary to create an application server cluster.

Use the following procedure to create an application server cluster. This procedure creates a cluster containing two application servers, each on a separate node. Run all commands as root user.

1. Logon to the test server.
2. Use a text editor to create a text file called createcluster.jy in the /tmp directory with the following content:

```
AdminTask.createCluster(['-clusterConfig [-clusterName TestCluster] -
replicationDomain [-createDomain true]]')
AdminTask.createClusterMember(['-clusterName TestCluster -memberConfig
[-memberNode TestNode01 -memberName TestServer01 -memberWeight 2 -
replicatorEntry true] -firstMember [-templateName default -nodeGroup
DefaultNodeGroup -coreGroup DefaultCoreGroup]]')
AdminTask.createClusterMember(['-clusterName TestCluster -memberConfig
[-memberNode TestNode02 -memberName TestServer02 -memberWeight 2 -
replicatorEntry true]]')
AdminConfig.list('ServerCluster', AdminConfig.getid( '/Cell:TestCell01/'))
AdminConfig.save()
Sync = AdminControl.completeObjectName('type=NodeSync,node=TestNode01,*')
AdminControl.invoke(Sync, 'sync')
Sync = AdminControl.completeObjectName('type=NodeSync,node=TestNode02,*')
AdminControl.invoke(Sync, 'sync')
```

**Note:** This script assumes that you have two nodes named TestNode01 and TestNode02 in a cell named TestCell01.

3. Enter the following commands to run the script created in the previous step to create the cluster:

```
cd /usr/IBM/WebSphere70/AppServer/profiles/TestDmgr01/bin
```

```
./wsadmin.sh -lang jython -f /tmp/createcluster.jy
```

4. Once the script has completed, log on to the administration console and verify that the cluster was created.
5. Optional: Deploy any applications that you wish to test in the new environment using your normal deployment methods.

An application server cluster called TestCluster consisting of two application servers is created. The application servers are called TestServer01 and TestServer02 and are located on nodes TestNode01 and TestNode02 respectively.

## Migrating a WAS 6.1 test environment to WAS 7.0

This topic describes the migration of an existing WAS 6.1 ND clustered test environment to WAS 7.0.

There are several benefits to migrating a test environment before migrating your production environment. These include:

- Allows you to gain experience of the migration process
- Gives you a migrated environment to test that your applications and configuration continue to work correctly
- Helps you to determine whether any problems are introduced by the migration process

## Migrating WAS 6.1 to 7.0

---

This section describes the steps necessary to migrate a test environment from WAS 6.1 to 7.0.

The high level steps to migrate from WAS 6.1 to WAS 7.0 are:

1. Install WAS 7.0.
2. Update WAS 7.0 to the latest maintenance level.
3. Migrate the WAS configuration from 6.1 to 7.0.

### Installing WAS 7.0

This task describes the steps necessary to install WebSphere Application Server Network Deployment 7.0.

Use the following procedure to install the WebSphere Application Server Network Deployment product. This procedure installs the product ready for use in a new environment or allows migration from a previous version. Run all commands as root user.

1. Obtain the install media appropriate for your platform. For example: Download part number **C1G2QML** for 64-bit AIX.
2. Enter the following command to create a temporary directory to hold the install media:

```
mkdir /tmp/was7
```

3. Copy the software image to the temporary directory.
4. Enter the following commands to extract the install media into the temporary directory:

```
gzip -d C1G2QML.tar.gz
```

```
tar -xvf C1G2QML.tar
```

5. Use a text editor to create a response file called `responsefile.txt` in `/tmp/was7/WAS` with the following content:

```
-OPT silentInstallLicenseAcceptance="true";
-OPT installType="installNew";
-OPT profileType="none";
-OPT feature="noFeature";
-OPT installLocation="/usr/IBM/WebSphere70/AppServer"
```

6. Enter the following commands to start the installer in silent mode using the response file created in the previous step:

```
cd /tmp/was7/WAS
```

```
./install -options "/tmp/was7/WAS/responsefile.txt" -silent
```

**Note:** If serious problems are encountered that prevent the product from installing, the errors are logged in the `/waslogs` directory. Review the files in this directory, correct any problems (for example insufficient space on target file systems), and retry the installation.

- a) Verify that the installation succeeded using the `installver.sh` command. This command verifies that the product files were installed as expected and reports on any variations.

```
/usr/IBM/WebSphere70/AppServer/bin/installver.sh
```

7. If the installation files are no longer needed, they can be deleted by running the following command:

```
rm -r /tmp/was
```

The product files are installed into the `/usr/IBM/WebSphere70/AppServer` directory on the target machine.

## Updating WAS 7.0 to the latest maintenance level

This topic describes the high level steps necessary to update WebSphere Application Server Network Deployment 7.0 to the latest maintenance level.

The high level steps required to update the WAS ND 7.0 to the latest maintenance level are:

1. Install the latest version of the IBM Update Installer for WebSphere Software.
2. Apply the latest WAS fix pack.
3. Apply the latest WAS SDK fix pack.

Details for each of these steps are provided in the sections that follow.

### Installing the IBM Update Installer for WebSphere Software

This task describes the steps necessary to install the IBM Update Installer for WebSphere Software. This software is used to install maintenance and updates to WebSphere Application Server.

Use the following procedure to install the IBM Update Installer for WebSphere Software. Run all commands as root user.

1. Obtain the install file appropriate for your platform from <http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg24020212>.

a) For example: download 7.0.0.1-WS-UPDI-AixPPC64.tar.gz for 64 bit AIX.

2. Enter the following command to create a temporary directory to hold the install media:

```
mkdir /tmp/updi
```

3. Copy the software image to the temporary directory.

4. Enter the following commands to extract the install media to the temporary directory:

```
gzip -d 7.0.0.1-WS-UPDI-AixPPC64.tar.gz
```

```
tar -xvf 7.0.0.1-WS-UPDI-AixPPC64.tar
```

5. Use a text editor to create a response file called `responsefile.txt` in `/tmp/updi/UpdateInstaller` with the following content:

```
-OPT silentInstallLicenseAcceptance="true"  
-OPT installLocation="/usr/IBM/WebSphere70/UpdateInstaller"
```

6. Enter the following commands to start the installer in silent mode using the response file created in the previous step:

```
cd /tmp/updi/UpdateInstaller
```

```
./install -options "/tmp/updi/UpdateInstaller/responsefile.txt" -silent
```

**Note:** If serious problems are encountered that prevent the product from installing, the errors are logged in the `/updi/logs` directory. Review the files in this directory, correct any problems (for example insufficient space on target file systems), and retry the installation.

- a) Verify that the installation succeeded using the `installver.sh` command. This command verifies that the product files were installed as expected and reports on any variations.

```
/usr/IBM/WebSphere70/UpdateInstaller/bin/installver.sh
```

7. If the installation files are no longer needed, they can be deleted by running the following command:

```
rm -r /tmp/updi
```

The software is installed into the `/usr/IBM/WebSphere70/UpdateInstaller` directory on the target machine.

### Applying a fix pack to WAS 7.0

This task describes the steps necessary to use the IBM Update Installer for WebSphere Software to update WebSphere Application Server Network Deployment 7.0 with the latest recommended fix pack.

The list of recommended fixes for all versions of WebSphere can be found at the following location:

<http://www-01.ibm.com/support/docview.wss?uid=swg27004980>

At the time of writing, the most recent available fix pack for WebSphere Application Server 7.0 is Fix Pack

1. The instructions below apply to that fix pack.

Use the following procedure to apply a fix pack to WAS 7.0. Run all commands as root user.

1. Obtain the appropriate fix pack file for your platform.
  - a) For example, download 7.0.0-WS-WAS-AixPPC64-FP0000001.pak for 64 bit AIX.
2. Copy the fix pack file to the maintenance directory under the Update Installer directory which is `/usr/IBM/WebSphere70/UpdateInstaller/maintenance`
3. Use a text editor to create a response file called `install7.0.0.1.txt` in the directory `/usr/IBM/WebSphere70/UpdateInstaller/responsefiles` with the following content:

```
-W product.location="/usr/IBM/WebSphere70/AppServer"
-W maintenance.package="/usr/IBM/WebSphere70/UpdateInstaller/
maintenance/7.0.0-WS-WAS-AixPPC64-FP0000001.pak"
-W update.type="install"
```

4. Enter the following commands to start the installer in silent mode using the response file created in the previous step:

```
cd /usr/IBM/WebSphere70/UpdateInstaller
```

```
./update.sh -options "/usr/IBM/WebSphere70/UpdateInstaller/responsefiles/
install7.0.0.1.txt" -silent
```

5. Enter the following command to run the VersionInfo tool:

```
/usr/IBM/WebSphere70/AppServer/bin/versionInfo.sh
```

- a) Ensure that the update was applied by checking that the Version field correctly reflects the fix pack you applied, in this case 7.0.0.1. The output should be similar to the following:

```
WVER0010I: Copyright (c) IBM Corporation 2002, 2005, 2008; All rights
reserved.
WVER0012I: VersionInfo reporter version 1.15.1.26, dated 8/9/08
```

```
-----
IBM WebSphere Application Server Product Installation Status Report
-----
```

```
Report at date and time 06 January 2009 14:45:41 GMT
```

```
Installation
-----
```

```
Product Directory /usr/IBM/WebSphere70/AppServer
Version Directory /usr/IBM/WebSphere70/AppServer/properties/version
DTD Directory /usr/IBM/WebSphere70/AppServer/properties/version/dtd
Log Directory /usr/IBM/WebSphere70/AppServer/logs
Backup Directory /usr/IBM/WebSphere70/AppServer/properties/version/nif/
backup
TMP Directory /tmp
```

```
Product List
-----
```

```
ND installed
```

```
Installed Product
-----
```

```
Name IBM WebSphere Application Server - ND
Version 7.0.0.1
ID ND
Build Level cf010845.08
Build Date 11/12/08
Architecture PPC64
```

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

WAS ND 7.0 is upgraded to the latest maintenance level on the target machine.

### Applying a fix pack to the WAS 7.0 SDK

This task describes how to use the IBM Update Installer for WebSphere Software to update WebSphere Application Server 7.0 Software Development Kit (SDK) with the latest recommended fixes.

The list of recommended fixes for all versions of WebSphere can be found at the following location:

<http://www-01.ibm.com/support/docview.wss?uid=swg27004980>

At the time of writing, the most recent available Java SDK fix pack for WebSphere Application Server 7.0 is Fix Pack 1. The instructions below are applicable for that fix pack.

Use the following procedure to apply a fix pack to WAS 7.0 SDK. Run all commands as root user.

1. Obtain the appropriate fix pack file for your platform.
  - a) For example download 7.0.0-WS-WASSDK-AixPPC64-FP0000001.pak for 64 bit AIX
2. Copy the fix pack file to the maintenance directory under the Update Installer directory which is `/usr/IBM/WebSphere70/maintenance`
3. Using a text editor create a response file called `installSDK7.0.0.1.txt` in the directory `/usr/IBM/WebSphere70/UpdateInstaller/responsefiles` with the following content:

```
-W product.location="/usr/IBM/WebSphere70/AppServer"
-W maintenance.package="/usr/IBM/WebSphere70/UpdateInstaller/
maintenance/7.0.0-WS-WASSDK-AixPPC64-FP0000001.pak"
-W update.type="install"
```

4. Run the update program using the response file created in the previous step:

```
cd /usr/IBM/WebSphere70/UpdateInstaller
```

```
./update.sh -options "/usr/IBM/WebSphere70/UpdateInstaller/responsefiles/
installSDK7.0.0.1.txt" -silent
```

5. Verify that the update was applied using the version information command:

```
/usr/IBM/WebSphere70/AppServer/java/bin/java -fullversion
```

- a) The output should be similar to the following. Ensure that the Version field correctly reflects the Java SDK fix pack you applied. For example, the following message will be displayed for the 7.0.0.1 Java SDK

```
java full version "JRE 1.6.0 IBM AIX build pap6460sr3-20081106_07 (SR3)"
```

WAS 7.0 SDK is upgraded to the latest maintenance level.

## Migrating the WAS configuration

This task describes the steps necessary to migrate a WAS Network Deployment 6.1 clustered test environment to WAS 7.0.

Before starting the migration ensure that you have performed the following tasks:

- Read the planning section and make any changes required for your applications to run in WAS 7.0.
- Install WAS 7.0 and apply the latest maintenance level on each machine hosting a managed node in the WAS 6.1 test environment to be migrated.
- Verify that the WAS 6.1 environment to be migrated is stable and prevent any administration activity or configuration changes being made until the migration is complete. The WAS 6.1 system can be left up and running until key stages within the migration when you will be required to stop various processes.
- Unconfigure any IBM Tivoli Composite Application Manager (ITCAM) for WebSphere Data Collectors that you have configured for your WAS application servers. For details see Appendix "ITCAM Considerations." You will need to reconfigure these following the migration. Backup the existing WAS 6.1 profiles using the `backupConfig` command. This is good practice, so that in the event of any problems, this configuration can be easily restored using the `restoreConfig` command.

**Note:** The migration process does not change the existing WAS 6.1 configuration (other than to disable the deployment manager and node agents to prevent them from being started accidentally). The WASPreUpgrade step backs up the 6.1 configuration, but the format of this backup does not support an easy restoration of the configuration.

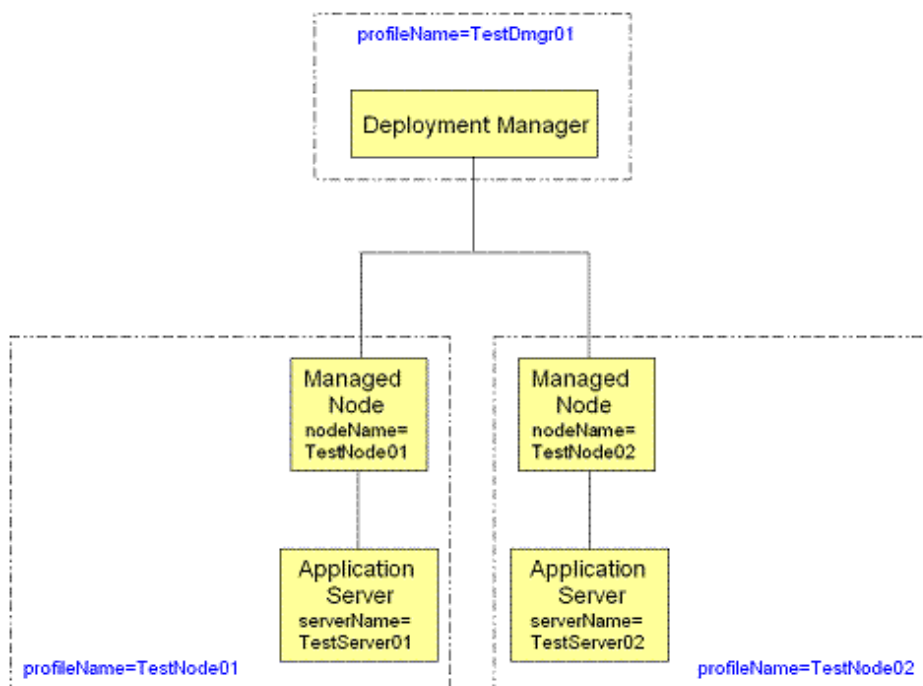
- Collect the following information:

- WAS 6.1 configuration cell name
- WAS 6.1 deployment manager's profile name and node name
- For each WAS 6.1 managed node: profile name and node name.
- The userids under which the WAS processes run.

This procedure will migrate the deployment manager profile, followed by each managed node profile in turn.

**Important:** The deployment manager must always be at the highest release and fix level within a cell for it to manage all nodes in the cell. The Version 7.0 deployment manager can manage Version 5.1.x, Version 6.x, and Version 7.0 nodes. This allows a cell to be upgraded to a new release one node at a time, with minimal impact to the applications that are running within the cell.

The following diagram describes the WAS configuration in the test environment to be migrated.



**Figure 3: WAS configuration in test environment**

These instructions make the following assumptions:

WAS 6.1 has been installed into the directory: /usr/IBM/WebSphere61/AppServer.

WAS 7.0 has been installed into the directory: /usr/IBM/WebSphere70/AppServer.

Use the following procedure to migrate a WAS 6.1 clustered test environment to WAS 7.0. Run all commands as root user, unless otherwise specified.

1. Migrate the deployment manager.
  - a) Create a WAS 7.0 deployment manager profile.

This profile will become the target profile during the WASPostUpgrade migration step.

**Important:** The *cellName* must match the value used for the WAS 6.1 deployment manager's profile and the *nodeName* must either match the value used for the 6.1 deployment manager or be unique within the cell.

Example: To create a WAS 7.0 deployment manager profile called TestDmgr01 with administration security enabled enter the following commands:

```
cd /usr/IBM/WebSphere70/AppServer/bin
```

```
./manageprofiles.sh -create -templatePath /usr/IBM/WebSphere70/
AppServer/profileTemplates/management -profileName TestDmgr01 -
profilePath /usr/IBM/WebSphere70/AppServer/profiles/TestDmgr01/ -
enableAdminSecurity true -adminUserName wasadmin -adminPassword password
-cellName TestCell01 -nodeName TestDmgr01
```

- b) Run the WASPreUpgrade command for the deployment manager profile.

This saves the configuration of the previously installed version of the WAS profile into a migration-specific backup directory.

Example: To run the WAS 7.0 WASPreUpgrade command for the deployment manager profile enter the following commands:

```
cd /usr/IBM/WebSphere70/AppServer/bin
```

```
./WASPreUpgrade.sh /usr/IBM/WebSphere61/DmgrBackup /usr/IBM/WebSphere61/
AppServer -oldProfile TestDmgr01
```

**Note:** For full syntax of the WASPreUpgrade command see the WAS 7.0 InfoCenter.

- c) Increase SOAP connection timeouts if migrating a large ND configuration.

To avoid connection timeouts when migrating a large number of small applications or one large application increase the value of the SOAP request timeout property to be sufficient for the migration to complete (the value is in milliseconds).

Example: To increase the SOAP request timeout value from the WAS default of 180ms to 1800ms, edit the `soap.client.props` file to set `com.ibm.SOAP.requestTimeout=1800` in each of the following directories (make a note of the existing values first):

WAS 7.0 deployment manager profile's property directory: `/usr/IBM/WebSphere70/
AppServer/profiles/TestDmgr01/properties`

WAS 6.1 deployment manager profile's backupDirectory/profiles/profile\_name/properties
directory: `/usr/IBM/WebSphere61/DmgrBackup/profiles/TestDmgr01/properties`

- d) Run the WASPostUpgrade command for the deployment manager profile.

This retrieves the saved configuration that was created by the WASPreUpgrade command from the backupDirectory that you specified. The WASPostUpgrade script reads the configuration from this directory to migrate to the WAS 7.0 profile.

Example: To run the WAS 7.0 WASPostUpgrade command for the deployment manager profile enter the following commands:

```
cd /usr/IBM/WebSphere70/AppServer/bin
```

```
./WASPostUpgrade.sh /usr/IBM/WebSphere61/DmgrBackup -profileName
TestDmgr01 -oldProfile TestDmgr01 -username wasadmin -password password
-includeApps script -keepDmgrEnabled true
```

#### Notes:

1. The use of `-includeApps script` and `-keepDmgrEnabled false` are recommended to give more control over the migration, allowing you to deploy the applications to WAS 7.0 as a separate step. (The default options are `-includeApps true` and `-keepDmgrEnabled false`).
2. The default option `-scriptCompatibility true` is used to minimize the impact to existing administrative scripts. This provides a temporary transition until all of the nodes in the

environment are at WAS 7.0, at which time you should perform the actions required to convert the configuration to use WAS 7.0 settings.

3. For full syntax of the `WASPostUpgrade` command see the WAS 7.0 InfoCenter.

- e) Install the enterprise applications.

Ensure all the enterprise applications deployed to the WAS 6.1 configuration are deployed to the WAS 7.0 configuration.

If the default option `-includeApps true` is specified on the `WASPostUpgrade` command then all applications are automatically migrated as part of that step and you need take no further action. If the option `-includeApps script` is specified on the `WASPostUpgrade` command, then at this time the applications can be deployed to the WAS 7.0 deployment manager. The `WASPostUpgrade` command copies the enterprise application ear files from WAS 6.1 into the WAS 7.0 deployment manager profile's `installableApps` directory and creates `wsadmin jython` scripts to install the applications in the deployment manager profile's `backup` directory. A script called `install_all_apps.jy` is created, along with `jython` scripts to install each individual application.

Example: To run the generated `wsadmin jython` script to install all migrated applications enter the following commands:

```
cd /usr/IBM/WebSphere70/AppServer/profiles/TestDmgr01/bin
./wsadmin.sh -f /usr/IBM/WebSphere61/DmgrBackup/install_all_apps.jy -
conntype NONE -lang jython
```

**Important:** If you have a large number of applications, you may wish to reorganize and combine the install application scripts created by the `WASPostUpgrade` command for better applications installation efficiency, rather than running the single `install_all_apps.jy` script. However, you **MUST** install all applications that were deployed in your WAS 6.1 environment to your WAS 7.0 deployment manager **BEFORE** attempting to start the WAS 7.0 deployment manager

- f) Stop the WAS 6.1 deployment manager and disable it.

**Important:** The WAS 6.1 deployment manager **MUST** be stopped before starting the WAS 7.0 deployment manager.

If default option `-KeepDmgrEnabled false` is specified on the `WASPostUpgrade` command then the WAS 6.1 deployment manager is automatically stopped and disabled and you need take no further action.

If option `-KeepDmgrEnabled true` is specified on the `WASPostUpgrade` command, then at this time you must manually stop the WAS 6.1 deployment manager. It is recommended that you also disable the WAS 6.1 deployment manager to prevent an accidental start.

Example: To disabling the WAS 6.1 deployment manager by renaming its `serverindex.xml` file enter the following commands:

```
cd /usr/IBM/WebSphere61/AppServer/profiles/TestDmgr01/config/cells/
TestCell01/nodes/TestDmgr01/
mv serverindex.xml servindex.xml_disabled
```

- g) If migrating a WAS configuration running as a non-root user: change the ownership of the deployment manager profile's directory structure.

The userid under which the WAS deployment manager process runs should be the owner of the deployment manager profile's directory structure, to ensure it has sufficient access permissions to all the files it requires to run successfully.

Example: To change the ownership of the deployment manager profile's directory structure where the deployment manager process runs as user=wasuser defined in group=wasgroup, enter the following commands:

```
cd /usr/IBM/WebSphere70/AppServer/profiles
chown -R wasuser:wasgroup TestDmgr01
```

- h) Start the WAS 7.0 deployment manager.

**Important:** If migrating a WAS configuration running as a non-root user: ensure you are logged on as the non-root user before issuing the startManager command, otherwise the deployment manager will fail to start due to insufficient file permissions.

Example: To start the WAS 7.0 deployment manager issue the following command:

```
cd /usr/IBM/WebSphere70/AppServer/profiles/TestDmgr01/bin
./startManager.sh
```

- i) Check that the WAS 7.0 deployment manager is working correctly.

Ensure the deployment manager starts cleanly (check SystemOut.log and SystemErr.log).

Ensure you can log in to administration console and that Welcome page displays version as 7.0.x.x.

Ensure administration console shows node agents are started, nodes are synchronized, application servers are started and their versions are displayed as 6.1.x.

- j) Reset the SOAP connection timeouts.

If you increased the SOAP connection timeouts during the migration of the deployment manager you can reset them now.

Example: Edit the `soap.client.props` file to set `com.ibm.SOAP.requestTimeout` back to its original value in each of the following directories:

WAS 7.0 deployment manager profile's property directory: `/usr/IBM/WebSphere70/AppServer/profiles/TestDmgr01/properties`

WAS 6.1 deployment manager profile's backupDirectory/profiles/profile\_name/properties directory: `/usr/IBM/WebSphere61/DmgrBackup/profiles/TestDmgr01/properties`

A WAS 7.0 deployment manager is running with WAS 6.1 node agents and application servers.

## 2. Migrate ONE of the managed nodes:

- a) Create a WAS 7.0 managed node profile

This profile will become the target profile during the WASPostUpgrade migration step.

### Important:

The `cellName` and `nodeName` must match the values for the corresponding WAS 6.1 managed node profile.

Do NOT federate the custom profiles into the cell, this will be done during the WASPostUpgrade step.

Example: To create a WAS 7.0 managed node profile called TestNode01 enter the following commands:

```
cd /usr/IBM/WebSphere70/AppServer/bin

./manageprofiles.sh -create -templatePath /usr/IBM/WebSphere70/AppServer/profileTemplates/management -profileName TestDmgr01 -
profilePath /usr/IBM/WebSphere70/AppServer/profiles/TestDmgr01/ -
enableAdminSecurity true -adminUserName wasadmin -adminPassword password
-cellName TestCell01 -nodeName TestDmgr01
```

- b) Run the WASPreUpgrade command for deployment manager profile.

This saves the configuration of the previously installed version of the WAS profile into a migration-specific backup directory.

Example: To run the WAS 7.0 WASPreUpgrade commands for the deployment manager profile enter the following commands:

```
cd /usr/IBM/WebSphere70/AppServer/bin

./manageprofiles.sh -create -templatePath /usr/IBM/WebSphere70/
AppServer/profileTemplates/managed -profileName TestNode01 -
profilePath /usr/IBM/WebSphere70/AppServer/profiles/TestNode01 -nodeName
TestNode01 -cellName TestNode01 -federateLater true
```

**Note:** For full syntax of the WASPreUpgrade command see the WAS 7.0 InfoCenter.

- c) Run the WASPreUpgrade command for the managed node profile.

Example: to run the WAS 7.0 WASPreUpgrade commands for a managed node profile enter the following commands:

```
cd /usr/IBM/WebSphere70/AppServer/bin

./WASPreUpgrade.sh /usr/IBM/WebSphere61/Node01Backup /usr/IBM/
WebSphere61/AppServer -oldProfile TestNode01
```

**Note:** For full syntax of the WASPreUpgrade command see the WAS 7.0 InfoCenter.

- d) Run the WASPostUpgrade command for the managed node profile.

This retrieves the saved configuration that was created by the WASPreUpgrade command from the backupDirectory that you specified. The WASPostUpgrade script reads the configuration from this directory to migrate to the WAS 7.0 profile.

Example: To run the WAS 7.0 WASPostUpgrade commands for a managed node profile enter the following commands:

```
cd /usr/IBM/WebSphere70/AppServer/bin

./WASPostUpgrade.sh /usr/IBM/WebSphere61/Node01Backup -profileName
TestNode01 -oldProfile TestNode01 -username wasadmin -password password
```

**Note:**

The WASPostUpgrade script stops and disables the WAS 6.1 node agent but leaves any associated application servers running. The WASPostUpgrade script initiates a full synchronization from the migrated WAS 7.0 node agent to the deployment manager.

For full syntax of the WASPostUpgrade command see the WAS 7.0 InfoCenter.

- e) Stop all the WAS 6.1 application servers associated with the migrated managed node profile.

**Important:** All WAS 6.1 application servers associated with a specific managed node MUST be stopped before the WAS 7.0 version of that node agent is started.

Example: To stop the WAS 6.1 application server TestServer01 on node TestNode01 from the command line issue the following commands:

```
cd /usr/IBM/WebSphere70/AppServer/profiles/TestNode01/bin
./stopServer.sh TestServer01
```

- f) If migrating a WAS configuration running as a non-root user: change the ownership of the managed node profile's directory structure.

The userid under which the node agent and application server processes run should be the owner of the managed node profile's directory structure, to ensure they have sufficient access permissions to all the files they require to run successfully.

Example: To change the ownership of the managed node profile's directory structure where the node agent and application server processes runs as user=wasuser defined in group=wasgroup, enter the following commands:

```
cd /usr/IBM/WebSphere70/AppServer/profiles
chown -R wasuser:wasgroup TestNode01
```

- g) Start the WAS 7.0 managed node's node agent.

Start the WAS 7.0 node agent for the migrated managed node in preparation for starting the migrated WAS 7.0 application servers.

Example: To start the WAS 7.0 node agent for node TestNode01 from the command line issue the following commands:

```
cd /usr/IBM/WebSphere70/AppServer/profiles/TestNode01/bin
./startNode.sh
```

- h) If the WAS configuration integrates with WebSphere MQ (WMQ): configure the WebSphere MQ JMS Provider definitions to reference the WMQ libraries.

If applications deployed to the WAS 7.0 configuration access WebSphere MQ using Bindings mode, it is necessary to configure the IBM WebSphere MQ JMS Provider's Native library path parameter to reference the location of the WebSphere MQ libraries. For details see the Appendix "WMQ Considerations."

- i) Start the WAS 7.0 application servers associated with the migrated managed node profile.

Use the WAS 7.0 administration Console to start the WAS 7.0 application server(s).

Example: To start a WAS 7.0 application server called TestServer01 using the WAS 7.0 administration console complete the following steps:

- Log in to the WAS 7.0 administration console.
- Expand `Servers`
- Expand `Server Types`
- Expand `WebSphere application servers`
- Select `TestServer01`.
- Click `Start`
- Wait for the server startup to complete.

- j) Check that the WAS 7.0 node agent and application server(s) started successfully.

WAS 7.0 deployment manager is running with one WAS 7.0 node agent and its application server(s) and remaining node agent(s) and application server(s) running at WAS 6.1.

### 3. Migrate the remaining managed nodes.

Repeat the steps performed for migrating the first managed node for each of the remaining nodes as appropriate for your environment and requirements.

### 4. Reconfigure ITCAM for WebSphere Data Collectors.

If your WAS 6.1 environment had ITCAM for WebSphere Data Collectors configured, you can reconfigure them now to work with the migrated WAS 7.0 environment. For details see Appendix "ITCAM Considerations."

### 5. Convert the WAS 7.0 configuration to use version 7.0 settings.

Once all nodes in the configuration have been migrated to WAS 7.0 you should ensure that the following configuration settings are changed to use WAS 7.0 definitions:

- Transport
- ProcessDef

- SSL configuration

If you selected the option `-scriptCompatibility=false` on the `WASPostUpgrade` command, then this will have been performed automatically during the migration and you need take no further action. If you accepted the default option `-scriptCompatibility=true` on the `WASPostUpgrade` command, then these configuration settings were kept at their WAS 6.1 definitions to minimize the impact on any administrative scripts and you should use the following procedure to update your WAS 7.0 configuration:

- Review your administration scripts and update them (where applicable) to use the WAS 7.0 settings: Channels instead of Transport, ProcessDefs instead of ProcessDef and use of version 7.0 SSL.
- Run the `convertScriptCompatibility` command to convert WAS 7.0 configuration to use the WAS 7.0 settings:  
Example: To run the `convertScriptCompatibility` command for all nodes and application servers issue the following commands:

```
cd /usr/IBM/WebSphere70/AppServer/profiles/TestDmgr01/bin
./convertScriptCompatibility.sh
```

**Note:**

This command **MUST** be run from the deployment manager profile's bin directory.

The default options backs up the configuration first and convert all application servers on all nodes. Options are available to convert specific nodes or application servers for more granular control.

For full syntax of the `convertScriptCompatibility` command see the WAS 7.0 InfoCenter.

- Perform a full resynchronization of the deployment manager's configuration with each federated node to produce a consistent configuration.

A running WAS 7.0 clustered environment is created containing the applications and configuration migrated from WAS 6.1.

After performing the migration you may wish to migrate your WAS 7.0 configuration to use the five character message ids.

In WAS 6.1 and 7.0 the format of message ids changed from four characters to five characters. The four character message ids have been deprecated but are maintained for users with a dependency on them. If you had not previously migrated to the five character message ids in WAS 6.1 the message ids will still be four characters in your migrated WAS 7.0 configuration and the following message will appear in the SystemOut.log of the deployment manager, node agents and application servers: TRAS0111I: The message ids that are in use are deprecated. Provided that you have no dependency on the four char message ids (for example, by log parsers or monitoring tools) then you can migrate to the five character message ids as follows

To change the message ids in the JVM logs:

- Set the JVM property `com.ibm.websphere.logging.messageId.version = 6` for each WAS 7.0 java process (deployment manager, node agent and application servers).
- Use a text editor to create a text file called `setJVMLogsMessageIds.jy` in the `/tmp` directory with the following content:

```
ls = java.lang.System.getProperty("line.separator")
cfgJvmList = AdminConfig.list("JavaVirtualMachine").split(ls)
for i in range(len(cfgJvmList)):
    cfgJvm = cfgJvmList[i]
    print "Set message id value for JVM "+cfgJvm.split("|")[0]
```

```
AdminConfig.create('Property', cfgJvm,
[[ 'name', 'com.ibm.websphere.logging.messageId.version'], ['value', '6'],
[ 'required', 'false']])
AdminConfig.save()
```

- Enter the following commands to run the wsadmin jython script:

```
cd /usr/IBM/WebSphere70/AppServer/profiles/TestDmgr01/bin
./wsadmin.sh -lang jython -f /tmp/setJVMLogsMessageIds.jy
```

To change the message ids in the startManager, stopManager, startNode, stopNode, startServer and stopServer log files:

- On UNIX:
  1. Append the following code to the end of the existingD\_ARGSparameter in the startManager.sh, stopManager.sh, startNode.sh, stopNode.sh, startServer.sh and stopServer.sh files in the install\_root/bin directory.
 

```
$DEBUG -Dcom.ibm.websphere.logging.messageId.version=6
```
  2. Restart the java processes for the changes to take effect.
- On Windows:
  1. Add the following line of code into the WRITE\_PROPERTIES\_FILE section of the startManager.bat, stopManager.bat, startNode.bat, stopNode.bat, startServer.bat and stopServer.bat files in the install\_root/bin directory:
 

```
>> %TMPJAVAPROFILE% echo com.ibm.websphere.logging.messageId.version=6
```
  2. Restart the java processes for the changes to take effect.

**Result:** All subsequent messages issued to the WAS 7.0 logs will contain five character message ids.

## Migrating a WAS 6.1 production environment to WAS 7.0

---

This topic describes the migration of an existing WAS 6.1 ND clustered production environment to WAS 7.0.

This project chose to migrate the entire WAS configuration first (deployment manager and managed nodes) and then migrate the Web servers (IBM HTTP Servers).

### Migrating WAS 6.1 to 7.0

---

This section describes the steps necessary to migrate a production environment from WAS 6.1 to 7.0.

The high level steps to migrate from WAS 6.1 to WAS 7.0 are:

- Install WAS 7.0.
- Update WAS 7.0 to the latest maintenance level.
- Migrate the WAS configuration from 6.1 to 7.0.

### Installing WAS 7.0

This task describes the steps necessary to install WebSphere Application Server Network Deployment 7.0.

Use the following procedure to install the WebSphere Application Server Network Deployment product.

This procedure installs the product ready for use in a new environment or allows migration from a previous version. Run all commands as root user.

1. Obtain the install media appropriate for your platform. For example: Download part number **C1G2QML** for 64-bit AIX.
2. Enter the following command to create a temporary directory to hold the install media:

```
mkdir /tmp/was7
```

3. Copy the software image to the temporary directory.
4. Enter the following commands to extract the install media into the temporary directory:

```
gzip -d C1G2QML.tar.gz
```

```
tar -xvf C1G2QML.tar
```

5. Use a text editor to create a response file called `responsefile.txt` in `/tmp/was7/WAS` with the following content:

```
-OPT silentInstallLicenseAcceptance="true";
-OPT installType="installNew";
-OPT profileType="none";
-OPT feature="noFeature";
-OPT installLocation="/usr/IBM/WebSphere70/AppServer"
```

6. Enter the following commands to start the installer in silent mode using the response file created in the previous step:

```
cd /tmp/was7/WAS
```

```
./install -options "/tmp/was7/WAS/responsefile.txt" -silent
```

**Note:** If serious problems are encountered that prevent the product from installing, the errors are logged in the `/waslogs` directory. Review the files in this directory, correct any problems (for example insufficient space on target file systems), and retry the installation.

- a) Verify that the installation succeeded using the `installver.sh` command. This command verifies that the product files were installed as expected and reports on any variations.

```
/usr/IBM/WebSphere70/AppServer/bin/installver.sh
```

7. If the installation files are no longer needed, they can be deleted by running the following command:

```
rm -r /tmp/was
```

The product files are installed into the `/usr/IBM/WebSphere70/AppServer` directory on the target machine.

## Updating WAS 7.0 to the latest maintenance level

This topic describes the high level steps necessary to update WebSphere Application Server Network Deployment 7.0 to the latest maintenance level.

The high level steps required to update the WAS ND 7.0 to the latest maintenance level are:

1. Install the latest version of the IBM Update Installer for WebSphere Software.
2. Apply the latest WAS fix pack.
3. Apply the latest WAS SDK fix pack.

Details for each of these steps are provided in the sections that follow.

### Installing the IBM Update Installer for WebSphere Software

This task describes the steps necessary to install the IBM Update Installer for WebSphere Software. This software is used to install maintenance and updates to WebSphere Application Server.

Use the following procedure to install the IBM Update Installer for WebSphere Software. Run all commands as root user.

1. Obtain the install file appropriate for your platform from <http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg24020212>.
  - a) For example: download `7.0.0.1-WS-UPDI-AixPPC64.tar.gz` for 64 bit AIX.
2. Enter the following command to create a temporary directory to hold the install media:

```
mkdir /tmp/updi
```

3. Copy the software image to the temporary directory.
4. Enter the following commands to extract the install media to the temporary directory:

```
gzip -d 7.0.0.1-WS-UPDI-AixPPC64.tar.gz
```

```
tar -xvf 7.0.0.1-WS-UPDI-AixPPC64.tar
```

5. Use a text editor to create a response file called `responsefile.txt` in `/tmp/updi/UpdateInstaller` with the following content:

```
-OPT silentInstallLicenseAcceptance="true"
-OPT installLocation="/usr/IBM/WebSphere70/UpdateInstaller"
```

6. Enter the following commands to start the installer in silent mode using the response file created in the previous step:

```
cd /tmp/updi/UpdateInstaller
./install -options "/tmp/updi/UpdateInstaller/responsefile.txt" -silent
```

**Note:** If serious problems are encountered that prevent the product from installing, the errors are logged in the `/updi/logs` directory. Review the files in this directory, correct any problems (for example insufficient space on target file systems), and retry the installation.

- a) Verify that the installation succeeded using the `installver.sh` command. This command verifies that the product files were installed as expected and reports on any variations.

```
/usr/IBM/WebSphere70/UpdateInstaller/bin/installver.sh
```

7. If the installation files are no longer needed, they can be deleted by running the following command:

```
rm -r /tmp/updi
```

The software is installed into the `/usr/IBM/WebSphere70/UpdateInstaller` directory on the target machine.

### Applying a fix pack to WAS 7.0

This task describes the steps necessary to use the IBM Update Installer for WebSphere Software to update WebSphere Application Server Network Deployment 7.0 with the latest recommended fix pack.

The list of recommended fixes for all versions of WebSphere can be found at the following location:

<http://www-01.ibm.com/support/docview.wss?uid=swg27004980>

At the time of writing, the most recent available fix pack for WebSphere Application Server 7.0 is Fix Pack 1. The instructions below apply to that fix pack.

Use the following procedure to apply a fix pack to WAS 7.0. Run all commands as root user.

1. Obtain the appropriate fix pack file for your platform.
  - a) For example, download `7.0.0-WS-WAS-AixPPC64-FP0000001.pak` for 64 bit AIX.
2. Copy the fix pack file to the maintenance directory under the Update Installer directory which is `/usr/IBM/WebSphere70/UpdateInstaller/maintenance`
3. Use a text editor to create a response file called `install7.0.0.1.txt` in the directory `/usr/IBM/WebSphere70/UpdateInstaller/responsefiles` with the following content:

```
-W product.location="/usr/IBM/WebSphere70/AppServer"
-W maintenance.package="/usr/IBM/WebSphere70/UpdateInstaller/
maintenance/7.0.0-WS-WAS-AixPPC64-FP0000001.pak"
-W update.type="install"
```

4. Enter the following commands to start the installer in silent mode using the response file created in the previous step:

```
cd /usr/IBM/WebSphere70/UpdateInstaller
./update.sh -options "/usr/IBM/WebSphere70/UpdateInstaller/responsefiles/
install7.0.0.1.txt" -silent
```

5. Enter the following command to run the VersionInfo tool:

```
/usr/IBM/WebSphere70/AppServer/bin/versionInfo.sh
```

- a) Ensure that the update was applied by checking that the Version field correctly reflects the fix pack you applied, in this case 7.0.0.1. The output should be similar to the following:

```
WVER0010I: Copyright (c) IBM Corporation 2002, 2005, 2008; All rights
reserved.
```

```

WVER0012I: VersionInfo reporter version 1.15.1.26, dated 8/9/08
-----
IBM WebSphere Application Server Product Installation Status Report
-----
Report at date and time 06 January 2009 14:45:41 GMT
Installation
-----
Product Directory /usr/IBM/WebSphere70/AppServer
Version Directory /usr/IBM/WebSphere70/AppServer/properties/version
DTD Directory /usr/IBM/WebSphere70/AppServer/properties/version/dtd
Log Directory /usr/IBM/WebSphere70/AppServer/logs
Backup Directory /usr/IBM/WebSphere70/AppServer/properties/version/nif/
backup
TMP Directory /tmp
Product List
-----
ND installed
Installed Product
-----
Name IBM WebSphere Application Server - ND
Version 7.0.0.1
ID ND
Build Level cf010845.08
Build Date 11/12/08
Architecture PPC64
-----
End Installation Status Report
-----

```

WAS ND 7.0 is upgraded to the latest maintenance level on the target machine.

### Applying a fix pack to the WAS 7.0 SDK

This task describes how to use the IBM Update Installer for WebSphere Software to update WebSphere Application Server 7.0 Software Development Kit (SDK) with the latest recommended fixes.

The list of recommended fixes for all versions of WebSphere can be found at the following location:

<http://www-01.ibm.com/support/docview.wss?uid=swg27004980>

At the time of writing, the most recent available Java SDK fix pack for WebSphere Application Server 7.0 is Fix Pack 1. The instructions below are applicable for that fix pack.

Use the following procedure to apply a fix pack to WAS 7.0 SDK. Run all commands as root user.

1. Obtain the appropriate fix pack file for your platform.
  - a) For example download 7.0.0-WS-WASSDK-AixPPC64-FP0000001.pak for 64 bit AIX
2. Copy the fix pack file to the maintenance directory under the Update Installer directory which is /usr/IBM/WebSphere70/maintenance
3. Using a text editor create a response file called installSDK7.0.0.1.txt in the directory /usr/IBM/WebSphere70/UpdateInstaller/responsefiles with the following content:

```

-W product.location="/usr/IBM/WebSphere70/AppServer"
-W maintenance.package="/usr/IBM/WebSphere70/UpdateInstaller/
maintenance/7.0.0-WS-WASSDK-AixPPC64-FP0000001.pak"
-W update.type="install"

```

4. Run the update program using the response file created in the previous step:

```
cd /usr/IBM/WebSphere70/UpdateInstaller
```

```
./update.sh -options "/usr/IBM/WebSphere70/UpdateInstaller/responsefiles/
installSDK7.0.0.1.txt" -silent
```

5. Verify that the update was applied using the version information command:

```
/usr/IBM/WebSphere70/AppServer/java/bin/java -fullversion
```

- a) The output should be similar to the following. Ensure that the Version field correctly reflects the Java SDK fix pack you applied. For example, the following message will be displayed for the 7.0.0.1 Java SDK

```
java full version "JRE 1.6.0 IBM AIX build pap6460sr3-20081106_07 (SR3)"
```

WAS 7.0 SDK is upgraded to the latest maintenance level.

## Migrating the WAS configuration

This task describes the steps necessary to migrate a WAS Network Deployment 6.1 clustered production environment to WAS 7.0.

Before starting the migration ensure that you have performed the following tasks:

- Proved the migration process and tested you applications and configuration in a WAS 7.0 test environment.
- Install WAS 7.0 and apply the latest maintenance level on each machine hosting a managed node in the WAS 6.1 production environment to be migrated.
- Verify that the WAS 6.1 environment to be migrated is stable and prevent any administration activity or configuration changes being made until the migration is complete. The WAS 6.1 system can be left up and running until key stages within the migration when you will be required to stop various processes.
- Unconfigure any IBM Tivoli Composite Application Manager (ITCAM) for WebSphere Data Collectors that you have configured for your WAS application servers. For details see Appendix "ITCAM Considerations". You will need to reconfigure these following the migration.
- Backup the existing WAS 6.1 profiles using the `backupConfig` command. This is good practice, so that in the event of any problems, this configuration can be easily restored using the `restoreConfig` command.

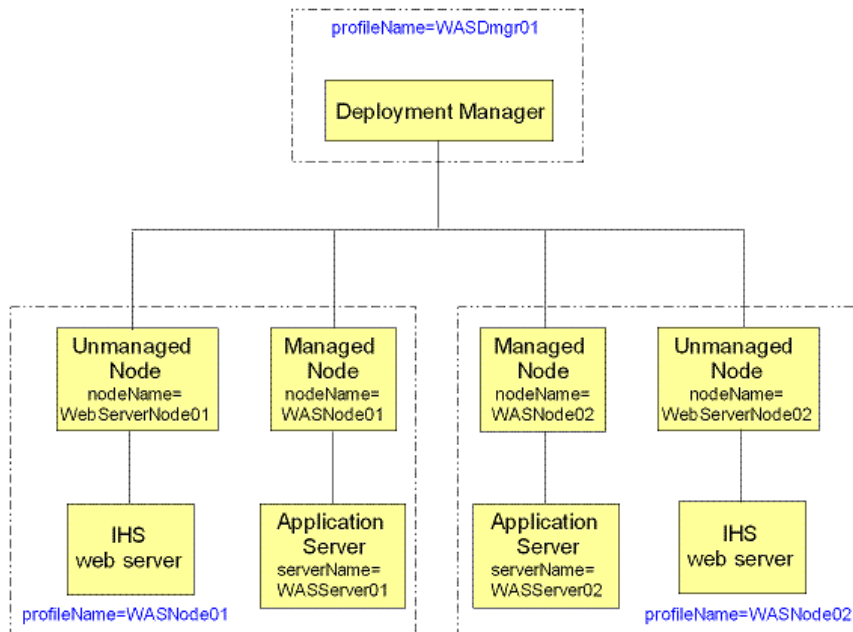
**Note:** The migration process does not change the existing WAS 6.1 configuration (other than to disable the deployment manager and node agents to prevent them from being started accidentally). The WASPreUpgrade step backs up the 6.1 configuration, but the format of this backup is not easy to restore from.

- Collect the following information:
  - WAS 6.1 configuration cell name.
  - WAS 6.1 deployment manager's profile name and node name.
  - For each WAS 6.1 managed node: profile name and node name.
  - The userids under which the WAS processes run.

This procedure will migrate the deployment manager profile, followed by each managed node profile in turn.

**Important:** The deployment manager must always be at the highest release and fix level within a cell for it to manage all nodes in the cell. The Version 7.0 deployment manager can manage Version 5.1.x, Version 6.x, and Version 7.0 nodes. This allows a cell to be upgraded to a new release one node at a time, with minimal impact to the applications that are running within the cell.

The following diagram describes the WAS configuration in the production environment to be migrated.



**Figure 4: WAS configuration in production environment**

These instructions make the following assumptions:

WAS 6.1 has been installed into the directory: /usr/IBM/WebSphere61/AppServer.

WAS 7.0 has been installed into the directory: /usr/IBM/WebSphere70/AppServer.

Use the following procedure to migrate a WAS 6.1 clustered production environment to WAS 7.0. Run all commands as root user, unless otherwise specified.

1. Migrate the deployment manager.
  - a) Create a WAS 7.0 deployment manager profile.

This profile will become the target profile during the WASPostUpgrade migration step.

**Important:** The *cellName* must match the value used for the WAS 6.1 deployment manager's profile and the *nodeName* must either match the value used for the 6.1 deployment manager or be unique within the cell.

Example: To create a WAS 7.0 deployment manager profile called WASDmgr01 with administration security enabled enter the following commands:

```
cd /usr/IBM/WebSphere70/AppServer/bin
```

```
./manageprofiles.sh -create -templatePath /usr/IBM/WebSphere70/
AppServer/profileTemplates/management -profileName WASDmgr01 -
profilePath /usr/IBM/WebSphere70/AppServer/profiles/WASDmgr01/ -
enableAdminSecurity true -adminUserName wasadmin -adminPassword password
-cellName TestCell01 -nodeName WASDmgr01
```

- b) Run the WASPreUpgrade command for the deployment manager profile.

This saves the configuration of the previously installed version of the WAS profile into a migration-specific backup directory.

Example: To run the WAS 7.0 WASPreUpgrade command for the deployment manager profile enter the following commands:

```
cd /usr/IBM/WebSphere70/AppServer/bin
./WASPreUpgrade.sh /usr/IBM/WebSphere61/DmgrBackup /usr/IBM/WebSphere61/
AppServer -oldProfile WASDmgr01
```

**Note:** For full syntax of the WASPreUpgrade command see the WAS 7.0 InfoCenter.

- c) Increase SOAP connection timeouts if migrating a large ND configuration.

To avoid connection timeouts when migrating a large number of small applications or one large application increase the value of the SOAP request timeout property to be sufficient for the migration to complete (the value is in milliseconds).

Example: To increase the SOAP request timeout value from the WAS default of 180ms to 1800ms, edit the `soap.client.props` file to set `com.ibm.SOAP.requestTimeout=1800` in each of the following directories (make a note of the existing values first):

```
WAS 7.0 deployment manager profile's property directory: /usr/IBM/WebSphere70/
AppServer/profiles/WASDmgr01/properties
WAS 6.1 deployment manager profile's backupDirectory/profiles/profile_name/properties
directory: /usr/IBM/WebSphere61/DmgrBackup/profiles/WASDmgr01/properties
```

- d) Run the WASPostUpgrade command for the deployment manager profile.

This retrieves the saved configuration that was created by the WASPreUpgrade command from the backupDirectory that you specified. The WASPostUpgrade script reads the configuration from this directory to migrate to the WAS 7.0 profile.

Example: To run the WAS 7.0 WASPostUpgrade command for the deployment manager profile enter the following commands:

```
cd /usr/IBM/WebSphere70/AppServer/bin
./WASPostUpgrade.sh /usr/IBM/WebSphere61/DmgrBackup -profileName
WASDmgr01 -oldProfile WASDmgr01 -username wasadmin -password password -
includeApps script -keepDmgrEnabled true
```

**Note:**

The use of `-includeApps script` and `-keepDmgrEnabled false` are recommended to give more control over the migration, allowing you to deploy the applications to WAS 7.0 as a separate step. (The default options are `-includeApps true` and `-keepDmgrEnabled false`).

The default option `-scriptCompatibility true` is used to minimize the impact to existing administrative scripts. This provides a temporary transition until all of the nodes in the environment are at WAS 7.0, at which time you should perform the actions required to convert the configuration to use WAS 7.0 settings.

For full syntax of the WASPostUpgrade command see the WAS 7.0 InfoCenter.

- e) Install the enterprise applications.

Ensure all the enterprise applications deployed to the WAS 6.1 configuration are deployed to the WAS 7.0 configuration.

If the default option `-includeApps true` is specified on the WASPostUpgrade command then all applications are automatically migrated as part of that step and you need take no further action.

If the option `-includeApps` script is specified on the `WASPostUpgrade` command then the applications can be deployed to the WAS 7.0 deployment manager now. The `WASPostUpgrade` command copies the enterprise application ear files from WAS 6.1 into the WAS 7.0 deployment manager profile's `installableApps` directory and creates `wsadmin jython` scripts to install the applications in the deployment manager profile's `backup` directory. A script called `install_all_apps.jy` is created, along with `jython` scripts to install each individual application.

Example: To run the generated `wsadmin jython` script to install all migrated applications enter the following commands:

```
cd /usr/IBM/WebSphere70/AppServer/profiles/WASDmgr01/bin
./wsadmin.sh -f /usr/IBM/WebSphere61/DmgrBackup/install_all_apps.jy -
conntype NONE -lang jython
```

**Important:** If you have a large number of applications, you may wish to reorganize and combine the install application scripts created by the `WASPostUpgrade` command for better applications installation efficiency, rather than running the single `install_all_apps.jy` script. However, you **MUST** install all applications that were deployed in your WAS 6.1 environment to your WAS 7.0 deployment manager **BEFORE** attempting to start the WAS 7.0 deployment manager.

- f) Stop the WAS 6.1 deployment manager and disable it.

**Important:** The WAS 6.1 deployment manager **MUST** be stopped before starting the WAS 7.0 deployment manager.

If default option `-KeepDmgrEnabled false` is specified on the `WASPostUpgrade` command then the WAS 6.1 deployment manager is automatically stopped and disabled and you need take no further action.

If option `-KeepDmgrEnabled true` is specified on the `WASPostUpgrade` command, then you must manually stop the WAS 6.1 deployment manager now. It is recommended that you also disable it to prevent it from being started accidentally.

Example: To disabling the WAS 6.1 deployment manager by renaming its `serverindex.xml` file enter the following commands:

```
cd /usr/IBM/WebSphere61/AppServer/profiles/WASDmgr01/config/cells/
TestCell01/nodes/WASDmgr01/
mv serverindex.xml servindex.xml_disabled
```

- g) If migrating a WAS configuration running as a non-root user: change the ownership of the deployment manager profile's directory structure.

The `userid` under which the WAS deployment manager process runs should be the owner of the deployment manager profile's directory structure, to ensure it has sufficient access permissions to all the files it requires to run successfully.

Example: To change the ownership of the deployment manager profile's directory structure where the deployment manager process runs as `user=wasuser` defined in `group=wasgroup`, enter the following commands:

```
cd /usr/IBM/WebSphere70/AppServer/profiles
chown -R wasuser:wasgroup WASDmgr01
```

- h) Start the WAS 7.0 deployment manager.

**Important:** If migrating a WAS configuration running as a non-root user: ensure you are logged on as the non-root user before issuing the `startManager` command, otherwise the deployment manager will fail to start due to insufficient file permissions.

Example: To start the WAS 7.0 deployment manager issue the following command:

```
cd /usr/IBM/WebSphere70/AppServer/profiles/WASDmgr01/bin
./startManager.sh
```

- i) Check that the WAS 7.0 deployment manager is working correctly

Ensure the deployment manager starts cleanly (check SystemOut.log and SystemErr.log)  
 Ensure you can log in to administration console and that Welcome page displays version as 7.0.x.x  
 Ensure administration console shows node agents are started, nodes are synchronized, application servers are started and their versions are displayed as 6.1.x.

- j) Reset the SOAP connection timeouts

If you increased the SOAP connection timeouts during the migration of the deployment manager you can reset them now.

Example: Edit the `soap.client.props` file to set `com.ibm.SOAP.requestTimeout` back to its original value in each of the following directories:

WAS 7.0 deployment manager profile's property directory: `/usr/IBM/WebSphere70/AppServer/profiles/WASDmgr01/properties`  
 WAS 6.1 deployment manager profile's backupDirectory/profiles/profile\_name/properties directory: `/usr/IBM/WebSphere61/DmgrBackup/profiles/WASDmgr01/properties`

**Result:** A WAS 7.0 deployment manager running with WAS 6.1 node agents and application servers.

## 2. Migrate ONE of the managed nodes

- a) Create a WAS 7.0 managed node profile

This profile will become the target profile during the WASPostUpgrade migration step.

### Important:

The `cellName` and `nodeName` must match the values for the corresponding WAS 6.1 managed node profile.

Do NOT federate the custom profiles into the cell, this will be done during the WASPostUpgrade step.

Example: To create a WAS 7.0 managed node profile called WASNode01 enter the following commands:

```
cd /usr/IBM/WebSphere70/AppServer/bin

./manageprofiles.sh -create -templatePath /usr/IBM/WebSphere70/AppServer/profileTemplates/management -profileName WASDmgr01 -profilePath /usr/IBM/WebSphere70/AppServer/profiles/WASDmgr01/ -enableAdminSecurity true -adminUserName wasadmin -adminPassword password -cellName TestCell01 -nodeName WASDmgr01
```

- b) Run the WASPreUpgrade command for deployment manager profile

This saves the configuration of the previously installed version of the WAS profile into a migration-specific backup directory

Example: To run the WAS 7.0 WASPreUpgrade commands for the deployment manager profile enter the following commands:

```
cd /usr/IBM/WebSphere70/AppServer/bin

./manageprofiles.sh -create -templatePath /usr/IBM/WebSphere70/AppServer/profileTemplates/managed -profileName WASNode01 -profilePath /usr/IBM/WebSphere70/AppServer/profiles/WASNode01 -nodeName WASNode01 -cellName WASNode01 -federateLater true
```

**Note:** For full syntax of the `WASPreUpgrade` command see the WAS 7.0 InfoCenter.

- c) Run the `WASPreUpgrade` command for the managed node profile  
 Example: to run the WAS 7.0 `WASPreUpgrade` commands for a managed node profile enter the following commands:

```
cd /usr/IBM/WebSphere70/AppServer/bin
```

```
./WASPreUpgrade.sh /usr/IBM/WebSphere61/Node01Backup /usr/IBM/
WebSphere61/AppServer -oldProfile WASNode01
```

**Note:** For full syntax of the `WASPreUpgrade` command see the WAS 7.0 InfoCenter.

- d) Run the `WASPostUpgrade` command for the managed node profile

This retrieves the saved configuration that was created by the `WASPreUpgrade` command from the `backupDirectory` that you specified. The `WASPostUpgrade` script reads the configuration from this directory to migrate to the WAS 7.0 profile.

Example: To run the WAS 7.0 `WASPostUpgrade` commands for a managed node profile enter the following commands:

```
cd /usr/IBM/WebSphere70/AppServer/bin
```

```
./WASPostUpgrade.sh /usr/IBM/WebSphere61/Node01Backup -profileName
WASNode01 -oldProfile WASNode01 -username wasadmin -password password
```

**Notes:**

1. The `WASPostUpgrade` script stops and disables the WAS 6.1 node agent but leaves any associated application servers running. It initiates a full synchronization from the migrated WAS 7.0 node agent to the deployment manager.
2. For full syntax of the `WASPostUpgrade` command see the WAS 7.0 InfoCenter.

- e) Stop all the WAS 6.1 application servers associated with the migrated managed node profile

**Important:** All WAS 6.1 application servers associated with a specific managed node **MUST** be stopped before the WAS 7.0 version of that node agent is started.

Example: To stop the WAS 6.1 application server `TestServer01` on node `WASNode01` from the command line issue the following commands:

```
cd /usr/IBM/WebSphere70/AppServer/profiles/WASNode01/bin
./stopServer.sh TestServer01
```

- f) If migrating a WAS configuration running as a non-root user: change the ownership of the managed node profile's directory structure.

The userid under which the node agent and application server processes run should be the owner of the managed node profile's directory structure, to ensure they have sufficient access permissions to all the files they requires to run successfully.

Example: To change the ownership of the managed node profile's directory structure where the node agent and application server processes runs as `user=wasuser` defined in `group=wasgroup`, enter the following commands:

```
cd /usr/IBM/WebSphere70/AppServer/profiles
chown -R wasuser:wasgroup WASNode01
```

- g) Start the WAS 7.0 managed node's node agent.

Start the WAS 7.0 node agent for the migrated managed node in preparation for starting the migrated WAS 7.0 application servers.

Example: To start the WAS 7.0 node agent for node WASNode01 from the command line issue the following commands:

```
cd /usr/IBM/WebSphere70/AppServer/profiles/WASNode01/bin
./startNode.sh
```

- h) If the WAS configuration integrates with WebSphere MQ (WMQ): configure the WebSphere MQ JMS Provider definitions to reference the WMQ libraries.

If applications deployed to the WAS 7.0 configuration access WebSphere MQ using Bindings mode, it is necessary to configure the IBM WebSphere MQ JMS Provider's Native library path parameter to reference the location of the WebSphere MQ libraries. For details see the Appendix "WMQ Considerations".

- i) Start the WAS 7.0 application servers associated with the migrated managed node profile.

Use the WAS 7.0 administration Console to start the WAS 7.0 application server(s).

Example: To start a WAS 7.0 application server called TestServer01 using the WAS 7.0 administration console complete the following steps:

- Log in to the WAS 7.0 administration console
  - Expand `Servers`
  - Expand `Server Types`
  - Expand `WebSphere application servers`
  - Select `TestServer01`
  - Click `Start`
  - Wait for the server startup to complete
- j) Check that the WAS 7.0 node agent and application server(s) started successfully.
- Result:** WAS 7.0 deployment manager running with one WAS 7.0 node agent and its application server(s) and remaining node agent(s) and application server(s) running at WAS 6.1.

### 3. Migrate the remaining managed nodes.

Repeat the steps performed for migrating the first managed node for each of the remaining nodes as appropriate for your environment and requirements.

### 4. Reconfigure ITCAM for WebSphere Data Collectors.

If your WAS 6.1 environment had ITCAM for WebSphere Data Collectors configured, you can reconfigure them now to work with the migrated WAS 7.0 environment. For details see Appendix "ITCAM Considerations".

### 5. Convert the WAS 7.0 configuration to use version 7.0 settings.

Once all nodes in the configuration have been migrated to WAS 7.0 you should ensure that the following configuration settings are changed to use WAS 7.0 definitions:

- Transport
- ProcessDef
- SSL configuration

If you selected the option `-scriptCompatibility=false` on the `WASPostUpgrade` command, then this will have been performed automatically during the migration and you need take no further action.

If you accepted the default option `-scriptCompatibility=true` on the `WASPostUpgrade` command, then these configuration settings were kept at their WAS 6.1 definitions to minimize the impact on any administrative scripts and you should use the following procedure to update your WAS 7.0 configuration:

- a) Review your administration scripts and update them (where applicable) to use the WAS 7.0 settings: Channels instead of Transport, ProcessDefs instead of ProcessDef and use of version 7.0 SSL.

- b) Run the `convertScriptCompatibility` command to convert WAS 7.0 configuration to use the WAS 7.0 settings:

Example: To run the `convertScriptCompatibility` command for all nodes and application servers issue the following commands:

```
cd /usr/IBM/WebSphere70/AppServer/profiles/WASDmgr01/bin
./convertScriptCompatibility.sh
```

**Note:**

This command **MUST** be run from the deployment manager profile's bin directory.

The default options backs up the configuration first and convert all application servers on all nodes. Options are available to convert specific nodes or application servers for more granular control.

For full syntax of the `convertScriptCompatibility` command see the WAS 7.0 InfoCenter.

- c) Perform a full resynchronization of the deployment manager's configuration with each federated node to produce a consistent configuration.

A running WAS 7.0 clustered environment is created containing the applications and configuration migrated from WAS 6.1.

After performing the migration you may wish to migrate your WAS 7.0 configuration to use the 5 character message ids.

In WAS 6.1 and 7.0 the format of message ids changed from four characters to five characters. The four character message ids have been deprecated but are maintained for users with a dependency on them. If you had not previously migrated to the 5 character message ids in WAS 6.1 the message ids will still be four characters in your migrated WAS 7.0 configuration and the following message will appear in the SystemOut.log of the deployment manager, node agents and application servers: TRAS01111: The message ids that are in use are deprecated. Provided that you have no dependency on the four char message ids (for example, by log parsers or monitoring tools) then you can migrate to the five character message ids as follows.

To change the message ids in the JVM logs:

- Set the JVM property `com.ibm.websphere.logging.messageId.version = 6` for each WAS 7.0 java process (deployment manager, node agent and application servers).
- Use a text editor to create a text file called `setJVMLogsMessageIds.jy` in the `/tmp` directory with the following content:

```
ls = java.lang.System.getProperty("line.separator")
cfgJvmList = AdminConfig.list("JavaVirtualMachine").split(ls)
for i in range(len(cfgJvmList)):
    cfgJvm = cfgJvmList[i]
    print "Set message id value for JVM "+cfgJvm.split("|")[0]
    AdminConfig.create('Property', cfgJvm,
    [['name', 'com.ibm.websphere.logging.messageId.version'], ['value', '6'],
    ['required', 'false']])
AdminConfig.save()
```

- Enter the following commands to run the `wsadmin` jython script:

```
cd /usr/IBM/WebSphere70/AppServer/profiles/WASDmgr01/bin
./wsadmin.sh -lang jython -f /tmp/setJVMLogsMessageIds.jy
```

To change the message ids in the `startManager`, `stopManager`, `startNode`, `stopNode`, `startServer` and `stopServer` log files:

- On UNIX:

1. Append the following code to the end of the existing `D_ARGSparameter` in the `startManager.sh`, `stopManager.sh`, `startNode.sh`, `stopNode.sh`, `startServer.sh` and `stopServer.sh` files in the `install_root/bin` directory.

```
$DEBUG -Dcom.ibm.websphere.logging.messageId.version=6
```

2. Restart the java processes for the changes to take effect.
- On Windows:
    1. Add the following line of code into the `WRITE_PROPERTIES_FILE` section of the `startManger.bat`, `stopManager.bat`, `startNode.bat`, `stopNode.bat`, `startServer.bat` and `stopServer.bat` files in the `install_root/bin` directory:
 

```
>> %TMPJAVAPROPPFILE% echo com.ibm.websphere.logging.messageId.version=6
```
    2. Restart the java processes for the changes to take effect.

**Result:** All subsequent messages issued to the WAS 7.0 logs will contain 5 character message ids.

## Migrating IBM HTTP Server (IHS) 6.1 to 7.0

This section describes the steps necessary to migrate IHS from 6.1 to 7.0 in an environment where IHS has been configured to integrate with WAS.

The high level steps to migrate from IHS 6.1 to 7.0 and integrate with WAS 7.0 are:

- Install IHS 7.0.
- Install the Web server plug-in.
- Configure the Web server plug-in to integrate with WAS 7.0.

### Installing IBM HTTP Server 7.0

This task describes the steps necessary to install IBM HTTP Server (IHS) 7.0.

Use the following procedure to install the IBM HTTP Server product. This procedure installs the product in a new directory structure, alongside any existing previous versions of IHS. Run all commands as root user.

1. Obtain the install file appropriate for your platform.
  - a) For example, download part number C1G2RML for 64 bit AIX: "IBM WebSphere Application Server Network Deployment V7.0 Supplements (1 of 2) for AIX on PowerPC 64-bit (Application Client, IBM HTTP Server, Web Server Plug-ins and Update Installer) Multilingual".
2. Enter the following command to create a temporary directory to hold the install media:

```
mkdir /tmp/ihs7
```

3. Copy the software image to the temporary directory.
4. Enter the following commands to extract the install media to the temporary directory:

```
gzip -d C1G2RML.tar.gz
```

```
tar -xvf C1G2RML.tar
```

5. Using a text editor, create a response file called `ihsresponsefile.txt` in `/tmp/ihs7/IHS` with the following content:

```
-OPT silentInstallLicenseAcceptance="true"
-OPT installLocation="/usr/IBM/HTTPServer70"
-OPT httpPort="80"
-OPT adminPort="8008"
-OPT createAdminAuth="false"
-OPT createAdminUserGroup="false"
-OPT installPlugin="false"
```

6. Enter the following commands to start the installer in silent mode using the response file created in the previous step:

```
cd /tmp/ihs7/IHS
```

```
./install -options "/tmp/ihs7/IHS/ihsresponsefile.txt" -silent
```

**Note:** If serious problems were encountered that prevented the product from installing, the errors are logged in the `/ihslogs` directory. Review the files in this directory, correct any problems (for example insufficient space on target file systems), and retry the installation.

- a) Verify that the installation succeeded using the `verifyinstallver.sh` command. This command verifies that the product files were installed as expected and reports on any variations.

```
/usr/IBM/HTTPServer70/bin/verifyinstallver.sh
```

7. Delete any temporary files and directories created in the previous steps if you will not use the installation files again.

```
rm -r /tmp/ihs7
```

IBM HTTP Server 7.0 is installed in `/usr/IBM/HTTPServer70`.

## Installing the IBM HTTP Server for WebSphere Application Server plug-in 7.0

This task describes the steps necessary to install the IBM HTTP Server for WebSphere Application Server plug-in 7.0.

Use the following procedure to install the IBM HTTP Server for WebSphere Application Server plug-in 7.0. Run all commands as root user.

1. Obtain the install file appropriate for your platform.
  - a) For example download part number C1G2RML for 64 bit AIX: "IBM WebSphere Application Server Network Deployment V7.0 Supplements (1 of 2) for AIX on PowerPC 64-bit (Application Client, IBM HTTP Server, Web Server Plug-ins and Update Installer) Multilingual."

2. Enter the following command to create a temporary directory to hold the install media:

```
mkdir /tmp/ihs7
```

3. Copy the software image to the temporary directory.

4. Enter the following commands to extract the install media to the temporary directory:

```
gzip -d C1G2RML.tar.gz
```

```
tar -xvf C1G2RML.tar
```

5. Using a text editor, create a response file called `plugin_responsefile.txt` in `/tmp/ihs7/plugin` with the following content:

```
-OPT silentInstallLicenseAcceptance="true"
-OPT installType="remote"
-OPT installLocation="/usr/IBM/HTTPServer70/Plugins"
-OPT wasExistingLocation="/usr/IBM/WebSphere70/AppServer"
-OPT webServerSelected="ihs"
-OPT ihsAdminPort="8008"
-OPT webServerConfigFile1="/usr/IBM/HTTPServer70/conf/httpd.conf"
-OPT webServerDefinition="WebServer01"
-OPT wasMachineHostName="fitmps02.hursley.ibm.com"
-OPT webServerHostName="fitmps02.hursley.ibm.com"
```

6. Enter the following commands to start the installer in silent mode using the response file created in the previous step:

```
cd /tmp/ihs7/plugin
```

```
./install -options "/tmp/ihs7/plugin/plugin_responsefile.txt" -silent
```

**Note:** If serious problems were encountered that prevented the product from installing, the errors are logged in the `/plglogs` directory. Review the files in this directory, correct any problems (for example insufficient space on target file systems), and retry the installation.

- a) Verify that the installation succeeded using the `verifyinstallver.sh` command. This command verifies that the product files were installed as expected and reports on any variations.

```
/usr/IBM/HTTPServer70/Plugins/bin/verifyinstallver.sh
```

7. Delete any temporary files and directories created in the previous steps if you will not use the installation files again.

```
rm -r /tmp/ihs7
```

IBM HTTP Server for WebSphere Application Server plug-in 7.0 is installed into /usr/IBM/HTTPServer70/Plugins.

## Configuring the web server plug-in.

This task describes the steps necessary to configure the IBM HTTP Server Web server to forward HTTP requests to WAS 7.0.

Before configuring the IBM HTTP Server to forward HTTP requests to WebSphere Application Server 7.0:

- Ensure IBM HTTP Server 7.0 has been installed and that it has been configured as required for your environment, for example, to enable SSL or customize listening ports.
- Ensure the IBM HTTP Server for WebSphere Application Server plug-in 7.0 has been installed.

Use the following procedure to configure IBM HTTP Server to forward HTTP requests to WAS 7.0 via the Web server plug-in. Run all commands as root user.

1. Stop IHS 6.1 server and administration server.

Example: To stop an IHS 6.1 server and administration server issue the following commands

```
cd /usr/IBM/HTTPServer61/bin
./apachectl -k stop
./apacheadmin stop
```

2. If the WAS configuration has not previously defined an IHS Web server, use the WAS 7.0 Administration Console to create an unmanaged node to host the IHS Web server. Follow these steps:

**Note:** If the WAS 7.0 configuration was migrated from a configuration which had previously defined an IHS Web server, an unmanaged node should already exist that can be reused for the IHS 7.0 web server.

- a) Log in to the WAS Admin Console.
- b) Expand System administration.
- c) Click Nodes.
- d) Click Add Node.
- e) Select Unmanaged node.
- f) Click Next.
- g) Fill in the information as follows:

```
Name: WebServerNode01
Host Name: webserverHostname
Platform Type: AIX
```

- h) Click OK.
- i) Click Save.

3. Use the following procedure to delete the current Web server:

- a) Log in to the WAS 7.0 Administration Console.
- b) Expand Servers.
- c) Select Web Servers.
- d) Select WebServer01
- e) Click Delete.
- f) Click Save.

4. Use the WAS 7.0 Administration Console to create a Web server definition for the IHS 7.0 Web server. Follow these steps:

- a) Log in to the WAS 7.0 Administration Console.
- b) Expand Servers.
- c) Click New Server.

- d) Select Web Server.
  - e) Click Next.
  - f) Fill in the information as follows:
 

```
Select node: WebServerNode01
Server name: WebServer01
Type: IBM HTTP Server
```
  - g) Click Next.
  - h) Click Next.
  - i) Fill in the information as follows:
 

```
Port: 80
Web Server installation location: /usr/IBM/HTTPServer70
Plug-in
installation location: /usr/IBM/HTTPServer70/Plugins
Application mapping to the WebServer: All
Administration Server Port: 8080
Username: httpadmin
Password: password
Confirm password: password
```
  - j) Click Next.
  - k) Click Finish.
  - l) Click Save.
  - m) Note: At this point the the deployment manager may attempt to propagate the plug-in configuration file, if so it will fail with errors similar to the following: PLGC0049E: The propagation of the plug-in configuration file failed for the Web server. WASCell01.WebServerNode01.WebServer01. Could not make a connection to the node agent or IBM HTTP Server administration server on node WebServerNode01.
5. Use the WAS 7.0 Administration Console to disable automatic propagation of the plug-in configuration file. This action should prevent these errors in future, for example, when a new application is deployed. Follow these steps:
    - a) Expand Server Types.
    - b) Click Web servers.
    - c) Click WebServer01.
    - d) Click Plug-in properties.
    - e) Uncheck Automatically propagate plug-in configuration file.
    - f) Click OK.
    - g) Click Save.
  6. Propagate the plugin-configuration file and SSL key files. Follow these steps:
    - a) Enter the following commands to copy the generated plug-in configuration file and SSL key files from the WAS 7.0 deployment manager to the IHS plug-in:
 

```
cd /usr/IBM/HTTPServer70/Plugins/config
mkdir WebServer01
cd WebServer01
scp root@WASDeploymentManagerHostname:/usr/IBM/WebSphere70/AppServer/
profiles/WASDmgr01/config/cells/WASCell01/nodes/WebServerNode01/servers/
WebServer01/plugin* .
```
    - b) Restart the HTTP Server for changes to take effect. Log in to the machine where IHS is installed and issue the following commands:
 

```
cd /usr/IBM/HTTPServer70/bin
./apachectl -k stop
./apachectl -k start
```

IHS is configured to forward HTTP requests to WAS 7.0 via the Web server plug-in.

## Considerations for integration with other products

This section describes other things you should consider before migrating your WAS configuration if it integrates with other products.

This project addressed products that are commonly configured to integrate with WAS:

- DB2
- WebSphere MQ (WMQ)
- IBM Tivoli Composite Application Manager for WebSphere (ITCAM for WebSphere)
- IBM Tivoli Directory Server

We considered the versions of these products that are supported with WAS 7.0. In addition, we considered the configuration or other user actions that are required for these products to continue to integrate successfully with WAS 7.0. For details, see the following sections:

- [DB2 considerations](#) on page 42
- [WebSphere MQ considerations](#) on page 43
- [ITCAM considerations](#) on page 44

### DB2 considerations

This section describes things you should consider when migrating a WAS environment configured to integrate with DB2 to WAS 7.0.

There are no special considerations for integrating with DB2 beyond ensuring that the DB2 server and any DB2 classes or native libraries referenced by WAS are at a supported version.

If your WAS application servers connect to a remote database server, then it is recommended that you ensure that the JDBC driver files installed on the application server machines are of the same version as the target database server.

**Table: DB2 versions supported with WAS 7.0**

Platform	Edition	Version
DB2 for Linux, UNIX and Windows	DB2 Connect for Linux, UNIX, and Windows DB2 Enterprise Server Edition for Linux, UNIX, and Windows DB2 Express for Linux, UNIX, and Windows DB2 Workgroup Server Edition for Linux, UNIX, and Windows	For all listed editions: FP8 V8.1 FP11a to FP15 V8.2 FP4a to V8.2 V9 to V9 FP4 V9.5 to V9.5 FP1
DB2 for z/OS		V8.1 V9 V9.1
DB2 for iSeries		V5.3 V5.4 V6.1

## JDBC drivers

The following Java Database Connectivity (JDBC) drivers are supported with WAS 7.0:

- DB2 Connect 8.2 FP8, 9.1 FP3 or 9.5 FP1
- DB2 Universal JDBC Driver 2.10.72 (type 2/4)
- DB2 Driver for JDBC and SQLJ 3.4.65 (type 2/4)
- IBM Dataserver Driver for JDBC and and SQLJ 3.51 (type 3) and 4.1 (type 4)

**Note:** For the latest information see the WebSphere Application Server System requirements page on the IBM Support Site and Technote #27013265 "Supported levels of DB2 for Linux, UNIX, and Windows for the IBM WebSphere Application Server V7.0 all editions."

## WebSphere MQ considerations

This section describes things you should consider when migrating a WAS environment configured to integrate with WebSphere MQ to WAS 7.0.

This section includes the following information:

- WMQ versions supported with WAS 7.0.
- Changes in how WAS 7.0 references WMQ libraries.
- Using 64 bit libraries.

### WebSphere MQ (WMQ) versions supported with WAS 7.0

The following table shows the WMQ versions that are currently supported with WAS 7.0:

WMQ version	BINDINGS mode	CLIENT mode
IBM WebSphere MQ V6.0.0.0 - V6.0.2.4	Not supported	Supported
IBM WebSphere MQ V6.0.2.5 or later	Supported WebSphere MQ V6.0.2.5 or later must be installed on the same machine as the application server	Supported
IBM WebSphere MQ V7	Supported WebSphere MQ V7 must be installed on the same machine as the application server	Supported

**Note:** For the latest information see the [System Requirements for WebSphere Application Server V7.0](#) on the IBM Support Site.

### Changes in how WAS 7.0 references WMQ libraries

The following information was obtained from [Technote #1316899 "WebSphere MQ messaging provider for WebSphere Application Server Version 7.0."](#) For full details, see the Technote.

```
"The WebSphere MQ messaging provider is installed as part of WebSphere Application Server V7. This includes a version of the WebSphere MQ V7 Resource Adapter, and the WebSphere MQ Extended Transactional Client functionality, which allows the queue manager to participate in XA transactions managed by the application server.
...
```

```
The WebSphere MQ messaging provider does not include the native libraries required to connect to a WebSphere MQ queue manager in BINDINGS mode. In order to use a BINDINGS mode connection, WebSphere MQ must be installed onto the same machine as the application server, and the Resource Adapter's native library path must be configured to point to the WebSphere MQ directory where these libraries are located.
...
```

```
Previous versions of WebSphere Application Server could be configured to use the WebSphere MQ classes for JMS located in an external WebSphere MQ installation to connect to a queue manager by setting the WebSphere variable MQ_INSTALL_ROOT. In WebSphere Application Server Version 7, MQ_INSTALL_ROOT is only used to locate native libraries, and is overridden by any native library path configured on the Resource Adapter."
```

## Actions required during the migration procedure

Use the following information during the migration procedure:

- If enterprise applications migrated to WAS 7.0 connect to WMQ in BINDINGS mode:
  - The MQ\_INSTALL\_ROOT environment variable value is not migrated. To continue to use BINDINGS mode to connect to WMQ you must manually specify the Native Libraries path property on the WebSphere MQ JMS Provider definition.
  - Instructions to set this value via the WAS Administration console are documented in “Configuring the WebSphere MQ messaging provider with native libraries information”
  - Alternatively this value can be set using wsadmin scripting. The following example jython set Native Library to /usr/mqm/java/lib64 at Node scope:

```
cellName = "myCell"
nodeName = "myNode"
nativeLibPathValue = "/usr/mqm/java/lib64"
scope = "/Cell:%s/Node:%s/" % (cellName,nodeName)
adapterName = "%sJ2CResourceAdapter:WebSphere MQ Resource Adapter/" % (scope)
adapterId = AdminConfig.getId(adapterName)
attr = "[-nativePath '"+nativeLibPathValue+"']"
AdminTask.manageWMQ(adapterId,attr)
AdminConfig.save()
```

**Note:** After setting or modifying the Native libraries path property on the WebSphere MQ JMS Provider definition, the associated application servers must be restarted twice to activate the change (this is due to the behaviour of the OSGi framework).

## Using 64 bit libraries in UNIX

If your WAS installation is 64 bit you should refer to the WMQ 64 bit libraries. In addition to specifying the path to the WMQ 64 bit library directory in the Native library path property of the JMS Provider definition, you may need to run the following command on each machine where 64 bit WMQ has been installed to unlink the 32 bit libraries:

- Log on as the root user and issue the following command:

```
dltmqlnk
```

**Note:** This command can be reversed using the mqcrtl link command. The dltmqlnk command will need to be re-run if any WMQ maintenance is applied. See WMQ infocentre for more information about these commands

## ITCAM considerations

---

This section describes the steps and considerations when migrating a WAS configuration that is monitored by IBM Tivoli Composite Application Manager (ITCAM) for WebSphere to WAS 7.0.

IBM Tivoli Composite Application Manager (ITCAM) for WebSphere is both a standalone product and an optional application included with WAS 7.0. The optional application is not compatible with IBM Tivoli Monitoring (ITM) and is therefore not used in our environment.

ITCAM for WebSphere is used to monitor the health of our WebSphere Application Server servers and applications. WAS 7.0 requires ITCAM for WebSphere V6.1 Fix Pack 4. As part of WAS 7.0 migration we applied Fix Pack 4 to our ITCAM for WebSphere, which initially ran at V6.1 Fix Pack 3 iFix 10.

## Updating ITCAM for WebSphere

This section describes the steps to update IBM Tivoli Composite Application Manager (ITCAM) for WebSphere.

Applying ITCAM for WebSphere V6.1 Fix Pack 4 involves updates to ITCAM for WebSphere Data Collectors, Tivoli Enterprise Monitoring Agents, Managing Server, and the IBM Tivoli Monitoring environment.

**Note:** Our environment does not include the Managing Server and therefore we did not install the associated Fix Pack. Additional steps required to update the Managing Server are not described in the following instructions.

**Note:** Our environment uses the Tivoli Enterprise Portal Browser Client and does not use the Tivoli Enterprise Portal Desktop Client. Additional steps required to update the Tivoli Enterprise Portal Desktop Client are not described in the following instructions

The high level steps required to update the ITCAM for WebSphere to the latest maintenance level are:

1. Apply application support to the IBM Tivoli Monitoring (ITM) environment.
2. Apply the latest Tivoli Enterprise Monitoring Agent (TEMA) fix pack.
3. Apply the latest Data Collector (DC) fix pack.

### Applying application support to the ITM for ITCAM for WebSphere

This section describes the steps necessary to apply ITCAM for WebSphere application support to ITM for the latest recommended fix pack.

The list of recommended fixes for all versions of ITCAM for WebSphere can be found at the following location:

<http://www-01.ibm.com/support/docview.wss?uid=swg21300120>

The TEMA fix pack is used to update application support in ITM. At the time of writing, the most recent available fix pack for ITCAM for WebSphere TEMA is V6.1 Fix Pack 4. The instructions below are applicable for that fix pack.

Use the following procedure to apply application support to the ITM components:

1. Obtain the appropriate fix pack file for your platform.
  - a) For example download 6.1.0-TIV-ITCAM\_WAS\_TEMA-AIX\_Solaris-FP004.tar for AIX
2. Enter the following command to create a temporary directory to hold the install media:
  - a) `mkdir /tmp/itcamfwas6104tema`
3. Copy the software image to the temporary directory
4. Enter the following command to extract the install media to the temporary directory:
 

```
tar -xvf 6.1.0-TIV-ITCAM_WAS_TEMA-AIX_Solaris-FP004.tar
```
5. Start the installer and follow the prompts:
 

```
./install.sh
```

  - a) Enter `/opt/IBM/ITM` as the name of the IBM Tivoli Monitoring directory.
  - b) Enter `y` to use the home directory `/opt/IBM/ITM` if it already exists.
  - c) If prompted that processes are currently running, continue with the installation and processes will be automatically stopped.
  - d) Select the option to install products to the local host.
  - e) Enter `1` to accept the licence agreement.
  - f) When prompted to select the product packages for the appropriate operating systems and component support categories select the number for Tivoli Enterprise Monitoring Server support.
  - g) Select the option to install the Monitoring Agent for WebSphere V06.10.04.00.
  - h) When prompted if you want to install additional products or product support packages, select the option for Yes.
  - i) When prompted to select the product packages for the appropriate operating systems and component support categories select the number for Tivoli Enterprise Portal Server support.
  - j) Select the option to install the Monitoring Agent for WebSphere V06.10.04.00.

- k) When prompted if you want to install additional products or product support packages, select the option for Yes.
- l) When prompted to select the product packages for the appropriate operating systems and component support categories select the number for Tivoli Enterprise Portal Browser Client support.
- m) Select the option to install the Monitoring Agent for WebSphere V06.10.04.00.
- n) When prompted if you want to install additional products or product support packages, select the option for No. The install will process the options selected, restart any processes that were stopped as part of the install, and end.

Application support for the ITCAM for WebSphere fix pack is added to the ITM environment.

### Applying a fix pack to ITCAM for WebSphere TEMAs

This section describes the steps necessary to apply the latest recommended fix pack to ITCAM for WebSphere TEMAs.

The list of recommended fixes for all versions of ITCAM for WebSphere can be found at the following location:

<http://www-01.ibm.com/support/docview.wss?uid=swg21300120>

At the time of writing, the most recent available fix pack for ITCAM for WebSphere TEMA is V6.1 Fix Pack 4. The instructions below are applicable for that fix pack.

Use the following procedure to apply the fix pack to the TEMAs:

1. Obtain the appropriate fix pack file for your platform.
  - a) For example download 6.1.0-TIV-ITCAM\_WAS\_TEMA-AIX\_Solaris-FP004.tar for AIX
2. Enter the following command to create a temporary directory to hold the install media:
  - a) `mkdir /tmp/itcamfwas6104tema`
3. Copy the software image to the temporary directory.
4. Enter the following command to extract the install media to the temporary directory:
 

```
tar -xvf 6.1.0-TIV-ITCAM_WAS_TEMA-AIX_Solaris-FP004.tar
```
5. Start the installer and follow the prompts:
 

```
./install.sh
```

  - a) Enter `/opt/IBM/ITM` as the name of the IBM Tivoli Monitoring directory.
  - b) Enter `y` to use the home directory `/opt/IBM/ITM` if it already exists.
  - c) If prompted that processes are currently running, continue with the installation and processes will be automatically stopped.
  - d) Select the option to install products to the local host.
  - e) Enter `1` to accept the licence agreement.
  - f) When prompted to select the product packages for the appropriate operating systems and component support categories select the number for AIX R5.3 (64 bit).
  - g) Select the option for "all of the above" to install both the Monitoring Agent for WebSphere V06.10.04.00 and the Tivoli Enterprise Services User Interface V06.10.07.03.
  - h) When prompted if you want to install additional products or product support packages, select the option for No. The install will process the options selected, restart any processes that were stopped as part of the install, and end
6. Repeat the steps above each TEMA.

The fix pack is applied to the TEMAs.

### Applying a fix pack to ITCAM for WebSphere Data Collectors

This section describes the steps necessary to apply the latest recommended fix pack to ITCAM for WebSphere Data Collectors (DCs).

The list of recommended fixes for all versions of ITCAM for WebSphere can be found at the following location:

<http://www-01.ibm.com/support/docview.wss?uid=swg21300120>

At the time of writing, the most recent available fix pack for ITCAM for WebSphere DC is V6.1 Fix Pack 4. The instructions below are applicable for that fix pack.

Use the following procedure to apply the fix pack to the DCs:

1. Obtain the appropriate fix pack file for your platform.
  - a) For example download 6.1.0-TIV-ITCAMfWAS\_MP-FP0004.tar for AIX
2. Enter the following command to create a temporary directory to hold the install media:
  - a) `mkdir /tmp/itcamfwas6104dc`
3. Copy the software image to the temporary directory.
4. Enter the following command to extract the install media to the temporary directory:
 

```
tar -xvf 6.1.0-TIV-ITCAMfWAS_MP-FP0004.tar
```
5. At this point there are two options. It is best practice to continue to the next step to unconfigure the existing DC and migrate WAS. Alternatively, skip to step 8. If you skip the unconfigure steps this can result in the data relating to the WAS 6.1 server DC persisting (for example: in TEP workspaces)
6. Navigate to the DC configuration directory:
 

```
cd /opt/IBM/itcam/WebSphere/DC/config_dc
```
7. Start the DC Configuration tool and follow the steps to unconfigure the DC:
 

```
./config_dc.sh
```

**Note:** The configure tool requires an X Windows display.

8. Set the Java home environment variable:
 

```
export JAVA_HOME=/opt/IBM/itcam/WebSphere/DC/_jvm/jre
```
9. Edit the `silentUpdate.properties` file to set the following attributes relating to the WebSphere Application Server Deployment Manager:
 

```
connection.host=fitmps01.hursley.ibm.com
connection.port=8879
connection.type=SOAP
connection.user=wasadmin
connection.password=password
```
10. Prepare the install:
 

```
./silentUpdate.sh -prepareInstall
```
11. Execute the install:
 

```
./silentUpdate.sh -install
```
12. Display the installed updates and check that the version is correct:
 

```
./silentUpdate.sh -displayInstalledUpdates
```
13. At this point you can migrate the WAS profile related to the DC's WAS server. Once you are satisfied that the WAS 7.0 is running correctly, continue to reconfigure the DC.
14. Navigate to the DC configuration directory:
 

```
cd /opt/IBM/itcam/WebSphere/DC/config_dc
```
15. Start the DC Configuration tool and follow the steps to configure the DC:
 

```
./config_dc.sh
```

**Note:** The configuration tool requires an X Windows display.

**Note:** In an ND environment, be sure to specify the Deployment Manager server (and not the application server) when prompted for the administration server details.

16. Repeat steps above for each DC.

The fix pack is applied to the DCs.

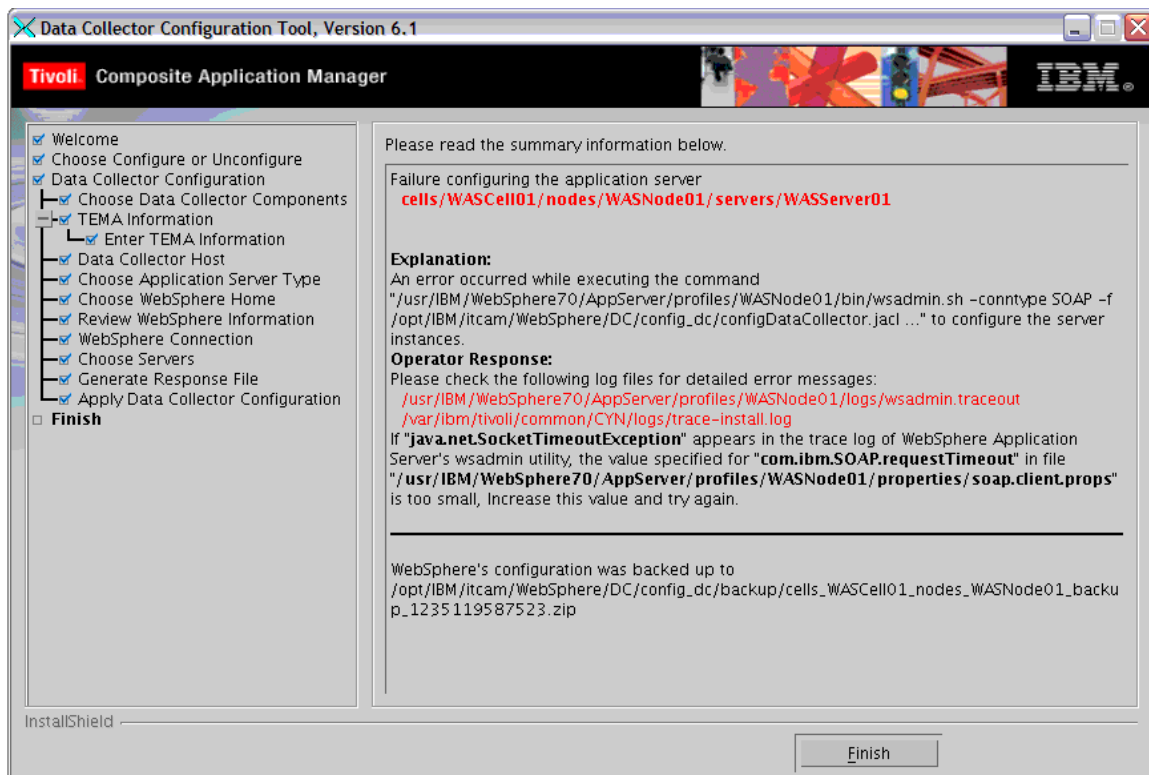
## Additional notes on updating ITCAM for WebSphere

This section provides additional guidance for updating ITCAM for WebSphere and describes associated problems and solutions.

1. Each fix pack comes with a readme which includes install instructions and guidance. It is prudent to read all readmes before applying the first fix pack.
2. The minimum prerequisite level of ITCAM for WebSphere for use with WAS 7.0 is ITCAM for WebSphere V6.1 fix pack 4. If you start a WAS 7.0 server with a DC running at a level below ITCAM for WebSphere V6.1 fix pack 4, the server can hang on start up with no obvious error messages, as shown in this startServer.log:

```
[19/02/09 14:04:21:287 GMT] 00000000 ManagerAdmin I   TRAS0017I: The startup trace state is *=info.
[19/02/09 14:04:21:470 GMT] 00000000 AdminTool   A   ADMU0128I: Starting tool with the WASNode01 profile
[19/02/09 14:04:21:477 GMT] 00000000 AdminTool   A   ADMU3100I: Reading configuration for server:
WASServer01
[19/02/09 14:04:22:175 GMT] 00000000 AdminTool   A   ADMU3200I: Server launched. Waiting for
initialization status.
```

3. It is a best practice to unconfigure the DC before you migrate to WAS 7.0 and then configure the DC after migrating to WAS 7.0. If you do not unconfigure the DC before migrating to WAS V7.0, you can simply configure the DC after migration to WAS 7.0. Failing to configure the DC correctly can cause problems, as follows:
  - Once ITCAM for WebSphere V6.1 fix pack 4 is applied and WAS 7.0 is installed, you must reconfigure the DC (i.e. using `/opt/IBM/itcam/WebSphere/DC/config_dc/config_dc.sh`). If you do not configure the DC before starting a WAS 7.0 server, the server will fail to start and create java core dumps (for example: `/usr/IBM/WebSphere70/AppServer/profiles/WASNode02/javacore.20090219.154406.774290.0003.txt`). The core dumps contain references to ITCAM (for example: `Module: /opt/IBM/itcam/WebSphere/DC/toolkit/lib/aix-64/libam_ibm_16.so`). In a WAS ND environment the nodeagent can cause the failing server to restart repeatedly causing a loop of server restarts.
  - If you configure the DC before updating to ITCAM for WebSphere V6.1 fix pack 4, the configuration tool will fail as shown below:



4. If you apply the ITCAM for WebSphere fix pack whilst ITCAM or ITM processes are running, the install process can automatically stop and restart processes (for example: TEMAs, TEMS, etc). If automatic

restart of processes occurs, be sure that the user you use apply the fix pack is intended and permitted to run the processes. Care should be taken to install and configure ITCAM for WebSphere DC with the correct user. If the DC runs as user wasuser (i.e. the WAS server user), you should install and configure the DC using user wasuser. Failing to use the correct user can cause problems, as follows:

- If the DC runs as user wasuser (i.e. the WAS server user), you install the DC using user root, and configure the DC as user wasuser, then you will see the following errors when starting config\_dc.sh:

```
wasuser$ ./config_dc.sh
Can not read or write file: /var/ibm/tivoli/common/CYN/FFDC/2009.02.19
Can not read or write file: /var/ibm/tivoli/common/CYN/FFDC
Can not read or write file: /var/ibm/tivoli/common/CYN
The log path is not writable: /var/ibm/tivoli/common
wasuser
```

- If the DC runs as user wasuser (i.e. the WAS server user), you install the DC using user root, and configure the DC as user root, and the configuration step is successful, when you start the WAS 7.0 server, the server can fail to start with the following errors:

- native\_stderr.log:

```
Exception in thread "Thread-2" java.security.AccessControlException: Access denied
(java.io.FilePermission /usr/IBM/WebSphere70/AppServer/profiles/WASNode02/temp/
osgi_instance_location/wasuser/2103141421 read)
    at java.security.AccessController.checkPermission(AccessController.java:108)
    at java.lang.SecurityManager.checkPermission(SecurityManager.java:532)
    at java.lang.SecurityManager.checkRead(SecurityManager.java:871)
    at java.io.File.isDirectory(File.java:764)
    at com.ibm.wsspi.bootstrap.WSPreLauncher.deleteEclipseInstanceArea(WSPreLauncher.java:576)
    at com.ibm.wsspi.bootstrap.WSPreLauncher.onExit(WSPreLauncher.java:567)
    at com.ibm.wsspi.bootstrap.WSPreLauncher$OnExit.run(WSPreLauncher.java:558)
    CCG0051E The log manager could not read the properties file /cyn-cclog.properties.
ITCAM Toolkit unable to load Properties - unable to continue Exception in thread "main"
java.security.AccessControlException: Access denied (java.util.PropertyPermission
eclipse.exitcode read)
    at java.security.AccessController.checkPermission(AccessController.java:108)
    at java.lang.SecurityManager.checkPermission(SecurityManager.java:532)
    at java.lang.SecurityManager.checkPropertyAccess(SecurityManager.java:1285)
    at java.lang.System.getProperty(System.java:378)
    at java.lang.System.getProperty(System.java:362)
    at org.eclipse.core.launcher.Main.run(Main.java:991)
    at com.ibm.wsspi.bootstrap.WSPreLauncher.launchEclipse(WSPreLauncher.java:330)
    at com.ibm.wsspi.bootstrap.WSPreLauncher.main(WSPreLauncher.java:108)
Exception in thread "Thread-2" java.security.AccessControlException: Access denied
(java.io.FilePermission /usr/IBM/WebSphere70/AppServer/profiles/WASNode02/temp/
osgi_instance_location/wasuser/2119638829 read)
    at java.security.AccessController.checkPermission(AccessController.java:108)
    at java.lang.SecurityManager.checkPermission(SecurityManager.java:532)
```

- native\_stdout.log:

```
2009-02-19 16:20:57.287+00:00 CCGBC0062W Class 'ccg_ffdc_filecopy_handler' must be defined
before object 'handler.ffdc.dc' can be created.
2009-02-19 16:20:57.288+00:00 CCGBC0062W Class 'ccg_ffdc_filecopy_handler' must be defined
before object 'handler.ffdc.dc' can be created.
2009-02-19 16:20:57.289+00:00 Error 0
2009-02-19 16:20:57.289+00:00 Error 0
2009-02-19 16:20:57.290+00:00 Error 0
2009-02-19 16:20:57.290+00:00 Error 0
2009-02-19 16:20:57.291+00:00 Error 0
2009-02-19 16:20:57.291+00:00 Error 0
2009-02-19 16:20:57.292+00:00 Error 0
2009-02-19 16:20:57.293+00:00 Error 0
2009-02-19 16:20:57.293+00:00 Error 0
2009-02-19 16:20:57.294+00:00 Error 0
2009-02-19 16:20:57.294+00:00 Error 0
2009-02-19 16:20:57.295+00:00 Error 0
2009-02-19 16:20:57.296+00:00 Error 0
2009-02-19 16:20:57.296+00:00 Error 0
2009-02-19 16:20:57.297+00:00 Error 0
2009-02-19 16:20:57.298+00:00 Error 0
```

Permission problems can be fixed by setting the correct file permission as follows:

```
chown -R wasuser:wasgroup /opt/IBM/itcam/WebSphere/DC/runtime
chown -R wasuser:wasgroup /var/ibm/tivoli/common
```

5. Once the WAS 7.0 server is up and running with the correctly configured DC, the Tivoli Enterprise Portal (TEP) will view the WAS 7.0 profile as a new profile (i.e. distinct from the old WAS 6.0 profile) and therefore TEP Managed System Lists and navigator views should be updated as required.

6. After making changes to the ITM environment any web browsers used to view the TEP should be stopped and restarted to avoid web browser caching problems.
7. After updating ITCAM for WebSphere, the TEP browser client became unresponsive when viewing detailed workspaces for WAS 7.0 servers. This was fixed by upgrading to the latest ITM maintenance level (V6.2.1 iFix 2).

## Rolling back to WAS 6.1

---

This section describes the process for rolling back a migrated environment to the previous level in the event of a problem.

It is recommended that a rollback to the WAS 6.1 configuration is performed in a test environment in order to ensure the steps required are understood and tested, should this be required in a production environment. Once a WAS configuration has been partially or fully migrated to WAS 7.0 the rollback procedure is not quite as simple as performing the same actions in reverse.

### Important:

- After a federated node has been migrated to WAS 7.0, do NOT restart the previous version (WAS 6.1) node agent while the WAS 7.0 deployment manager is running. Once a managed node has been migrated to WAS 7.0, the WAS 7.0 deployment manager expects it to be a WAS 7.0 node. If the version 6.1 node agent is started while the 7.0 deployment manager is running, the deployment manager will attempt to sync with it and could corrupt the 6.1 node configuration.
- The WAS 7.0 deployment manager and ALL WAS 7.0 node agents must be stopped before starting the WAS 6.1 deployment manager.
- The WAS 7.0 application servers must be stopped before their associated 6.1 node agent is started.
- Should the rollback procedure documented below fail, it will be necessary to restore the original 6.1 configuration from backups. In configurations where WAS is set to run as a non-root user, ensure that you set appropriate file permissions on the restored configurations. This is necessary because the zip file create by the backupConfig command does not retain file ownership information.

The following procedure keeps one application server running to maintain the service.

1. Stop WAS 7.0 deployment manager.
2. Stop ALL WAS 7.0 node agents.
3. Leave one WAS 7.0 application server running but stop all other application servers.  
This allows one application server to remain active to maintain a server. If this is not required, all WAS 7.0 application servers can be stopped at this point.
4. Enable WAS 6.1 deployment manager.  
The deployment manager was disabled during the migration, either manually or automatically. To enable it either run the migrationDisablementReversal.jacl that was created in the bin directory of the WAS 6.1 profile or manually replace the backed up serverindex.xml file.  
Example: manually enabling a WAS 6.1 deployment manager:
  1. `cd /usr/IBM/WebSphere61/AppServer/profiles/TestDmgr01/config/cells/TestCell01/nodes/TestDmgr01`
  2. `mv serverindex.xml_disabled serverindex.xml`
5. Start WAS 6.1 deployment manager.
6. Enable WAS 6.1 node agents for nodes that have no running application servers.  
The node agents were disabled automatically during the migration. To enable them either run the migrationDisablementReversal.jacl that was created in the bin directory of the WAS 6.1 profile or

manually replace the backed up `serverindex.xml` file. This must be done for each node agent that you wish to enable.

Example: manually enabling a WAS 6.1 node agent:

1. `cd /usr/IBM/WebSphere61/AppServer/profiles/TestNode01/config/cells/TestCell01/nodes/TestNode01`
2. `mv serverindex.xml_disabled serverindex.xml`

7. Start WAS 6.1 node agents for nodes that have no running application servers.
8. Start WAS 6.1 application servers for the nodes with running WAS 6.1 node agents.
9. Test applications work via the re-started WAS 6.1 application servers.
10. Stop the remaining WAS 7.0 application server.
11. Enable remaining WAS 7.0 node agent.
12. Start the remaining WAS 6.1 node agent.
13. Start the remaining WAS 6.1 application servers.
14. Test applications in fully restored WAS 6.1 configuration.

The WAS 6.1 system has been restored as the active running system and its configuration is as it was prior to any attempts to migrate to WAS 7.0.

## Troubleshooting

---

This section describes problems that were encountered during this project when attempting to migrate to WebSphere Application Server 7.0, and also notes some solutions to those problems. These problems relate to security and integration with additional products. They are documented here as an aid to other users who may also encountered them.

### Problems starting the migrated environment

---

This task describes a problem that may be encountered when issuing the `startManager` command for a deployment manager or `startNode` command for a node agent that has been migrated to WAS 7.0 and suggests a possible solution to that problem.

This task describes symptoms that may be encountered when issuing the command `startManager` to start the deployment manager or `startNode` to start a node agent for a WAS configuration running as a non-root user after migration to WAS 7.0.

The following messages will be displayed on the console:

```
ADMU0111E: Program exiting with error:
com.ibm.ws.process.exception.InvalidFileException: PROC0015E: The
process creation failed. Some files that are required by the
process or the process definition are inaccessible, such as standard
in, out, and error redirects. Verify that user [wasuser] has read
and write access to the following items: Working directory:
[/usr/IBM/WebSphere70/AppServer/profiles/TestDmgr01]. Standard
input: []. Standard output:
[/usr/IBM/WebSphere70/AppServer/profiles/TestDmgr01/logs/dmgr/
native_stdout.log].
Standard error:
[/usr/IBM/WebSphere70/AppServer/profiles/TestDmgr01/logs/dmgr/
native_stderr.log].
Empty [] indicates the value is not set and the operating system
defaults are used.
ADMU1211I: To obtain a full trace of the failure, use the -trace option.
ADMU0211I: Error details may be seen in the file:
```

```
/usr/IBM/WebSphere70/AppServer/profiles/TestDmgr01/logs/dmgr/startServer.log
```

This problem occurs when the start commands are issued from the root user or a userid other than that specified as the process owner in the WAS configuration. The first time a deployment manager, node agent or application server is started a new directory and log files are created in the profile's logs directory, this will be created with ownership permissions of the user that issues the start command. As the process starts it reaches a point where it determines the actual userid under which it should run, if this userid is different to the one that issued the start command it may not have authority to write to the log files.

To resolve this problem complete the following steps:

1. Change the ownership of all WAS 7.0 profile's directory structures to the user specified in the WAS process's configuration.

Example: To change ownership of a profile called TestDmgr01 to user=wasuser defined in group=wasgroup, issue the following commands:

```
cd /usr/IBM/WebSphere70/AppServer/profiles
```

```
chown -R wasuser:wasgroup TestDmgr01
```

2. Re-issue the `startManager` and `startNode` commands.

Deployment manager and node agents start successfully.

## Problems connecting to WebSphere MQ

---

This section describes problems that may be encountered when a WAS 7.0 application server attempts to connect to WebSphere MQ (WMQ) and suggests possible solutions to those problems.

### Problems after changing WMQ Native Library Path

This task describes a problem that may be encountered on the first restart of a WAS 7.0 application server after the Native Library Path property of a WebSphere MQ JMS Provider definition has been changed. The problems occur when the application server attempts to access the WMQ libraries and connect to the queue manager.

This task describes symptoms that may occur during start up of an application server when the server attempts to access the WMQ libraries.

The following messages will be displayed in the application server's SystemOut.log file:

```
WMSG1623E: The WebSphere MQ messaging provider installed at /usr/IBM/
WebSphere70/AppServer/installedConnectors/wmq.jmsra.rar has been updated
and an application server restart is required to pick up this update. The
WebSphere MQ messaging provider has been disabled.
```

```
Exception data follows: javax.naming.NamingException: WMSG2003E: It was not
possible to lookup the specified WebSphere MQ administered object because the
application server needs a restart to register an update to the WebSphere MQ
installation.
```

WAS 7.0 application server require two restarts to detect and load libraries specified by changes to the Native Library Path property of a WebSphere MQ JMS Provider.

To resolve this problem complete the following step:

Restart the application server.

Application server starts successfully and connects to the WebSphere MQ Queue Manager via its JMS Provider Queue Connection Factory definition.

## Problems connecting to WMQ when using 64 bit WAS

This task describes a problem that may occur during start up of an application server running in 64 bit mode when it attempts to access WMQ libraries.

This task describes symptoms that may occur during start up of an application server running in 64 bit mode when the server attempts to access WMQ libraries.

The following messages will be displayed in the application server's SystemOut.log file:

```
com.ibm.mq.jmqi.JmqiException: CC=2;RC=2495;AMQ8568: The native JNI library
'mqjbnj' was not found. [3=mqjbnj]
```

To resolve this problem complete the following steps:

1. Stop the application server.
2. Issue the following command from the root userid to unlink the WebSphere MQ 32 bit libraries:

```
dltmqlnk
```

**Note:** This allows the WAS application server to connect to the WMQ 64 bit libraries. The actions of the `dltmqlnk` command can be reversed using the `crtmqlnk` command, for details of these commands see the WebSphere MQ Information Center.

3. Start the application server.

Application server starts successfully and loads the WebSphere MQ 64 bit libraries.

## References

---

This section lists references that we found useful in performing the migrating to WAS 7.0 and which we recommend.

- WebSphere Application Server support site, "Knowledge Collection: Migration planning for WebSphere Application Server," located at <http://www-01.ibm.com/support/docview.wss?uid=swg27008724>
- DeveloperWorks article "A quick guide for migrating to WebSphere Application Server V7," located at [http://www.ibm.com/developerworks/websphere/library/techarticles/0812\\_luchini/0812\\_luchini.html](http://www.ibm.com/developerworks/websphere/library/techarticles/0812_luchini/0812_luchini.html)
- Software requirements:
  - WebSphere Application Server System requirements, located at <http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg27006921>
  - Technote #27013265 "Supported levels of DB2 for Linux, UNIX, and Windows for the IBM WebSphere Application Server V7.0 all editions," located at <http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg27013265>
  - Technote #1316899 "WebSphere MQ messaging provider for WebSphere Application Server Version 7.0," located at [http://www-01.ibm.com/support/docview.wss?rs=171&context=SSFKSJ&dc=DB540&uid=swg21316899&loc=en\\_US&cs=UTF-8&lang=en&rss=ct171websphere](http://www-01.ibm.com/support/docview.wss?rs=171&context=SSFKSJ&dc=DB540&uid=swg21316899&loc=en_US&cs=UTF-8&lang=en&rss=ct171websphere)
- WebSphere Application Server version 7.0 Information Center links:
  - "Migrating Network Deployment configurations for distributed operating systems," located at [http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.migration.nd.doc/info/ae/ae/tmig\\_nd.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.migration.nd.doc/info/ae/ae/tmig_nd.html)
  - "WASPreUpgrade command," located at [http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.migration.nd.doc/info/ae/ae/rmig\\_WASPreUpgrade.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.migration.nd.doc/info/ae/ae/rmig_WASPreUpgrade.html)
  - "WASPostUpgrade command," located at [http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.migration.nd.doc/info/ae/ae/rmig\\_WASPostUpgrade.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.migration.nd.doc/info/ae/ae/rmig_WASPostUpgrade.html)

- “convertScriptCompatibility command,” located at [http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.migration.nd.doc/info/ae/ae/xml\\_migratesecurity.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.migration.nd.doc/info/ae/ae/xml_migratesecurity.html)
- “Migrating administrative scripts from Version 6.x to Version 7.0,” located at [http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.migration.nd.doc/info/ae/ae/xml\\_migrate6to6.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.migration.nd.doc/info/ae/ae/xml_migrate6to6.html)
- “Changing the message IDs used in log files,” located at [http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/ttrb\\_chgdflmsg.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/ttrb_chgdflmsg.html)
- “Configuring the WebSphere MQ messaging provider with native libraries information,” located at [http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tmj\\_adm33.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tmj_adm33.html)
- Other documentation for the Mixed Platform Stack project:
  - "The Mixed Platform Stack Project: Deploying a secure SOA solution into z/OS and mixed z/OS and AIX environments v2.0," located at <http://www-03.ibm.com/support/techdocs/atmastr.nsf/WebIndex/WP101300>
  - "The Mixed Platform Stack Project: Deploying a Secure SOA Solution into z/OS and Mixed z/OS and AIX Environments" Redpiece, located at <http://www.redbooks.ibm.com/redpieces/abstracts/sg247724.html>

## Appendices

---

The following appendices are intended to support the information provided in the body of the document. They provide further details on topics which were covered by this project, but which were not our primary focus. The appendices include:

- Installing WAS 6.1
- Updating WAS 6.1

### Installing WAS 6.1

---

This section describes the steps necessary to install WebSphere Application Server Network Deployment 6.1.

Use the following procedure to install WebSphere Application Server ND 6.1 code in preparation for creating a new version 6.1 environment to test the migration process. Run all commands as root user.

1. Obtain the install media appropriate for your platform. For example download part number **C88TJML** for 64 bit AIX.
2. Enter the following command to create a temporary directory to hold the install media:

```
mkdir /tmp/was61
```

3. Copy the software image to the temporary directory.
4. Enter the following commands to extract the install media into the temporary directory:

```
gzip -d C88TJML.tar.gz
```

```
tar -xvf C88TJML.tar
```

5. Using a text editor, create a response file called `responsefile.txt` in `/tmp/was61/WAS` with the following content:

```
-silent
-OPT silentInstallLicenseAcceptance="true";
-OPT installType="installNew";
-OPT profileType="none";
-OPT feature="noFeature";
-OPT installLocation="/usr/IBM/WebSphere70/AppServer"
```

```
-OPT disableOSPrereqChecking="true"
```

**Note:** `disableOSPrereqChecking="true"` may be necessary if you are using a supported operating system with a later fix level than was available when WebSphere Application Server 6.1 was first released.

6. Enter the following commands to start the installer in silent mode using the response file created in the previous step:

```
cd /tmp/was61/WAS
```

```
./install -options "/tmp/was61/WAS/responsefile.txt" -silent
```

**Note:** If serious problems were encountered that prevented the product from installing the errors are logged in the `/waslogs` directory. Review the files in this directory, correct any problems (for example insufficient space on target file systems), and retry the installation.

- a) Verify that the installation succeeded using the `installver.sh` command. This command verifies that the product files were installed as expected and reports on any variations.

```
/usr/IBM/WebSphere61/AppServer/bin/installver.sh
```

7. Delete any temporary files and directories created in the previous steps if you will not use the installation files again.

```
rm -r /tmp/was61
```

WAS ND 6.1 is installed on the target machine.

## Updating WAS 6.1 to the latest maintenance level

This topic describes the steps necessary to update WebSphere Application Server Network Deployment 6.1 to the latest maintenance level.

The high level steps required to update the WAS ND 6.1 to the latest maintenance level are:

1. Install the latest version of the IBM Update Installer for WebSphere Software.
2. Apply the latest WAS fix pack.
3. Apply the latest WAS SDK fix pack.

Details for each of these steps are provided in the following sections.

### Installing the IBM Update Installer for WebSphere Software

The IBM Update Installer for WebSphere Software is used to apply maintenance and updates to WebSphere Application Server. Note that for WebSphere Application Server 6.1 you should use the most recent V7.0 Update Installer to install recent fix packs and fixes.

Use the following procedure to install the IBM Update Installer for WebSphere Software. Run all commands as root user.

1. Obtain the install file appropriate for your platform from <http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg24020212>.
  - a) For example download `7.0.0.1-WS-UPDI-AixPPC64.tar.gz` for 64 bit AIX.
2. Enter the following command to create a temporary directory to hold the install media:

```
mkdir /tmp/updi
```

3. Copy the software image to the temporary directory.
4. Enter the following commands to extract the install media to the temporary directory:

```
gzip -d 7.0.0.1-WS-UPDI-AixPPC64.tar.gz
```

```
tar -xvf 7.0.0.1-WS-UPDI-AixPPC64.tar
```

- Using a text editor, create a response file called `responsefile.txt` in `/tmp/updi/UpdateInstaller` with the following content:

```
-OPT silentInstallLicenseAcceptance="true"
-OPT installLocation="/usr/IBM/WebSphere61/UpdateInstaller"
```

- Enter the following commands to start the installer in silent mode using the response file created in the previous step:

```
cd /tmp/updi/UpdateInstaller
./install -options "/tmp/updi/UpdateInstaller/responsefile.txt" -silent
```

**Note:** If serious problems were encountered that prevented the product from installing, the errors are logged in the `/updi/logs` directory. Review the files in this directory, correct any problems (for example insufficient space on target file systems), and retry the installation.

- Verify that the installation succeeded using the `installver.sh` command. This command verifies that the product files were installed as expected and reports on any variations.

```
/usr/IBM/WebSphere61/UpdateInstaller/bin/installver.sh
```

- Delete any temporary files and directories created in the previous steps if you will not use the installation files again.

```
rm -r /tmp/updi
```

Latest version of the IBM Update Installer for WebSphere Software is installed.

## Applying a fix pack to WAS 6.1

Use the IBM Update Installer for WebSphere Software to update WebSphere Application Server 6.1 with the latest recommended fixes.

The list of recommended fixes for all versions of WebSphere can be found at the following location:

<http://www-01.ibm.com/support/docview.wss?uid=swg27004980>

At the time of writing, the most recent available fix pack for WebSphere Application Server 6.1 is Fix Pack 21. The instructions below are applicable for that fix pack.

Use the following procedure to apply a fix pack to WAS 6.1. Run all commands as root user.

- Obtain the appropriate fix pack file for your platform.
  - For example download `6.1.0-WS-WAS-AixPPC64-FP0000021.pak` for 64 bit AIX
- Copy the fix pack file to the maintenance directory under the Update Installer directory which is `/usr/IBM/WebSphere61/UpdateInstaller/maintenance`
- Using a text editor create a response file called `install6.1.0.21.txt` in the directory `/usr/IBM/WebSphere61/UpdateInstaller/responsefiles` with the following content:

```
-W product.location="/usr/IBM/WebSphere61/AppServer"
-W maintenance.package="/usr/IBM/WebSphere70/UpdateInstaller/
maintenance/6.1.0-WS-WAS-AixPPC64-FP0000021.pak"
-W update.type="install"
```

- Run the update program using the response file created in the previous step.

```
cd /usr/IBM/WebSphere61/UpdateInstaller
./update.sh -options "/usr/IBM/WebSphere70/UpdateInstaller/responsefiles/
install6.1.0.21.txt" -silent
```

- Verify that the update was applied using the version information command.

```
/usr/IBM/WebSphere61/AppServer/bin/versionInfo.sh
```

- The output should be similar to the following. Ensure that the Version field correctly reflects the fix pack you applied, in this case 7.0.0.1.

```
WVER0010I: Copyright (c) IBM Corporation 2002, 2005, 2008; All rights
reserved.
```

```

WVER0010I: Copyright (c) IBM Corporation 2002, 2005; All rights
reserved.
WVER0012I: VersionInfo reporter version 1.15.4.2, dated 6/5/08
-----
IBM WebSphere Application Server Product Installation Status Report
-----
Report at date and time 21 January 2009 11:45:06 GMT
Installation
-----
Product Directory      /usr/IBM/WebSphere61/AppServer
Version Directory     /usr/IBM/WebSphere61/AppServer/properties/
version
DTD Directory         /usr/IBM/WebSphere61/AppServer/properties/
version/dtd
Log Directory         /usr/IBM/WebSphere61/AppServer/logs
Backup Directory     /usr/IBM/WebSphere61/AppServer/properties/
version/nif/backup
TMP Directory         /tmp
Product List
-----
ND                    installed
Installed Product
-----
Name                  IBM WebSphere Application Server - ND
Version               6.1.0.21
ID                   ND
Build Level           cf210844.13
Build Date            11/6/08
-----
End Installation Status Report
-----

```

WAS 6.1 is upgraded to the latest maintenance level.

## Applying a fix pack to the WAS 6.1 SDK

Use the IBM Update Installer for WebSphere Software to update WebSphere Application Server 6.1 Software Development Kit (SDK) with the latest recommended fixes.

The list of recommended fixes for all versions of WebSphere can be found at the following location:

<http://www-01.ibm.com/support/docview.wss?uid=swg27004980>

At the time of writing, the most recent available Java SDK fix pack for WebSphere Application Server 6.1 is Fix Pack 21. The instructions below are applicable for that fix pack.

Use the following procedure to apply a fix pack to the WAS 6.1 SDK. Run all commands as root user.

1. Obtain the appropriate fix pack file for your platform.
  - a) For example, download 6.1.0-WS-WASSDK-AixPPC64-FP0000021.pak for 64-bit AIX.
2. Copy the fix pack file to the maintenance directory under the Update Installer directory which is `/usr/IBM/WebSphere61/maintenance`
3. Using a text editor create a response file called `installSDK6.1.0.21.txt` in the directory `/usr/IBM/WebSphere61/UpdateInstaller/responsefiles` with the following content:

```

-W product.location="/usr/IBM/WebSphere61/AppServer"
-W maintenance.package="/usr/IBM/WebSphere70/UpdateInstaller/
maintenance/6.1.0-WS-WASSDK-AixPPC64-FP0000021.pak"
-W update.type="install"

```

4. Run the update program using the response file created in the previous step.

```
cd /usr/IBM/WebSphere61/UpdateInstaller
```

```
./update.sh -options "/usr/IBM/WebSphere70/UpdateInstaller/responsefiles/
installSDK7.0.0.1.txt" -silent
```

5. Verify that the update was applied using the version information command.

```
/usr/IBM/WebSphere61/AppServer/java/bin/java -fullversion
```

- a) The output should be similar to the following. Ensure that the Version field correctly reflects the Java SDK fix pack you applied. For example, the following message will be displayed for the 7.0.0.1 Java SDK

```
java full version "J2RE 1.5.0 IBM AIX build pap64devifx-20080906 (SR8a + IZ29767 + IZ30684 + IZ31214 + IZ31213)"
```

WAS 6.1 SDK is upgraded to the latest maintenance level.

## Notices

---

The following notices apply to the content of this document.

### Disclaimer

Information in this document is provided 'AS IS' without warranty of any kind. Mention or reference to non-IBM products is for informational purposes only and does not constitute an endorsement of such products by IBM. Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

### Trademarks

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

- AIX®
- CICS®
- CICSplex®
- DB2®
- DB2® Universal Database
- IBM®
- IBM (logo)®
- Parallel Sysplex®
- RACF®
- Rational®
- Redbooks®
- Redbooks (logo)®
- System z
- Tivoli®
- WebSphere®
- z/OS®

The following terms are trademarks of other companies:

- Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- Linux is a trademark of Linus Torvalds in the United States, other countries, or both.
- Microsoft, Windows, and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.
- UNIX is a registered trademark of The Open Group in the United States and other countries.

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