



Migrating WebSphere Process Server



Migrating WebSphere Process Server

Note

Before using this information, be sure to read the general information in the Notices section at the end of this document.

27 June 2008

This edition applies to version 6, release 1, modification 2 of WebSphere Process Server for z/OS (product number 5655-N53) and to all subsequent releases and modifications until otherwise indicated in new editions.

To send us your comments about this document, send an e-mail message to doc-comments@us.ibm.com. We look forward to hearing from you.

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

© **Copyright International Business Machines Corporation 2006, 2008. All rights reserved.**

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

Contents

Chapter 1. Migrating from previous versions of WebSphere Process Server and WebSphere Enterprise Service Bus 1

Overview of migrating	1
Premigration considerations	2
How data is handled during migration from earlier versions	9
Configuration mapping during product-configuration migration	10
Migrating WebSphere applications.	13
Migrating stand-alone servers	14
Migrating a network deployment environment	17
Migrating a deployment manager	18
Migrating a managed node	20
Migrating Business Rules Manager in a network deployment environment	22
Migrating Business Rules Manager on the deployment target	22
Verifying migration.	23
Postmigration configuration checking.	24
Rolling back your environment.	25
Rolling back a deployment cell	25
Rolling back a managed node	28
Migration considerations for Business Process Choreographer	30
Troubleshooting version-to-version migration	32

Chapter 2. Migrating from previous WebSphere products 43

Migrating source artifacts from WebSphere Studio Application Developer Integration Edition	43
Migrating from WebSphere MQ Workflow	44

Chapter 3. Deprecated features 45

Deprecation list	45
Deprecated features in WebSphere Process Server version 6.1.2	45
Deprecated features in WebSphere Process server version 6.1.	46
Deprecated features in WebSphere Process Server version 6.0.2	49
Deprecated features in WebSphere Process Server version 6.0.1	51
Deprecated features in WebSphere Process Server version 6.0.	51
Deprecated features in WebSphere Business Integration Server Foundation version 5.1.1.	54
Deprecated features in WebSphere Business Integration Server Foundation version 5.1	54

Chapter 4. Troubleshooting migration 57

Troubleshooting version-to-version migration	57
--	----

Notices 67

Chapter 1. Migrating from previous versions of WebSphere Process Server and WebSphere Enterprise Service Bus

You can migrate your installed applications and configurations from prior versions of WebSphere® Process Server and WebSphere Enterprise Service Bus to WebSphere Process Server version 6.1.

Overview of migrating

Migrate from earlier versions of WebSphere Process Server and WebSphere Enterprise Service Bus.

Moving from one version of WebSphere Process Server to a newer release of WebSphere Process Server, or, in some cases, moving from a version of WebSphere Enterprise Service Bus to a higher release level of WebSphere Process Server is referred to as version-to-version migration. Version-to-version migration occurs when you install a new version of a product, such as WebSphere Process Server, and then copy relevant application and configuration data from the old installation to the new installation. With migration, the new version of the product is installed alongside the older product. Then, data is copied from the old version of the product to the new version of the product. Migrating is different from updating, in which out-of-date files or data of an existing installation are replaced with current information. Fix packs are an example of an update.

Migration must take place from an older version of WebSphere Process Server to a newer version that is running on the same operating system. You cannot migrate across different operating systems.

The following table shows the supported version-to-version migration scenarios for this release of WebSphere Process Server. You can migrate all of the products listed under "Currently installed version" to WebSphere Process Server version 6.1.

Currently installed version	New version
WebSphere Process Server 6.0.1.x	WebSphere Process Server 6.1
WebSphere Process Server 6.0.2.x	WebSphere Process Server 6.1
WebSphere Enterprise Service Bus 6.0.1.x	WebSphere Process Server 6.1
WebSphere Enterprise Service Bus 6.0.2.x	WebSphere Process Server 6.1

Why perform version-to-version migration?

WebSphere Process Server provides user application binary compatibility with prior versions. However, version-to-version migration allows you to preserve WebSphere Process Server configuration data in addition to your applications when moving to a newer version of WebSphere Process Server. Configuration of cells, clusters, servers and nodes are retained when you perform a version-to-version migration. If you did not perform this migration, and simply installed the new version of WebSphere Process Server, you would need to reconfigure your environment from scratch.

When upgrading from version 6.0.1 to version 6.0.2, in-place upgrades were available. Such an update preserves configuration data. Upgrading to version 6.1, however, requires a version-to-version migration.

Related concepts



Development and deployment version levels

Your decision about what version levels of WebSphere Process Server you need in your environment will depend on the version levels with which your applications were developed. Generally applications deployed in a previous version of WebSphere Process Server will run on the next available version of WebSphere Process Server.

Premigration considerations

Before you begin the process of migrating to a new version of WebSphere Process Server, you should be aware of these considerations.

The following rules and restrictions apply to migration and coexistence if you have WebSphere Process Server version 6.1 installed:

- You can migrate a version 6.0.x deployment manager to a version 6.1.x deployment manager only if they are at the same augmentation level.
- You cannot augment a version 6.1.x stand-alone server that has been migrated from a version 6.0.x stand-alone server.

You can create a new stand-alone profile in version 6.1.x and augment it.

- You cannot augment a version 6.1.x managed server that has been migrated from a 6.0.x managed server.

You can create a new profile in version 6.1.x, augment it, and then add the new node to a version 6.1.x cell that has an augmented deployment manager.

- You can have a mixed cell containing both augmented and unaugmented managed nodes as long as the cell's deployment manager has been augmented to the same augmentation level as the highest augmentation level of any of its managed nodes. For example, if the deployment manager is augmented for WebSphere Process Server, it can successfully manage nodes that have been augmented for WebSphere Process Server and WebSphere Application Server. However, a deployment manager that has only been augmented for WebSphere Application Server can only manage WebSphere Application Server nodes.

- If you have Business Process Choreographer installed, see Migration considerations for Business Process Choreographer.

- After you have installed WebSphere Process Server for z/OS® version 6.1, you might want to build a complete deployment cell configuration and make sure that it works properly before you attempt to migrate an existing cell or node.

This ensures that your system has all of the necessary prerequisites and supports the new level of WebSphere Process Server.

- High availability manager and core group functionality are included in WebSphere Process Server version 6.0 and later. See Core group migration considerations for core group configuration and topology considerations that might impact your migration from version 6.0.x to version 6.1.
- If you are migrating from WebSphere Process Server for z/OS version 6.0.1 to version 6.1 and security is enabled for version 6.0.1, you must upgrade to service level 6.0.2.12 or later before migrating to version 6.1.

If security is not enabled for WebSphere Process Server version 6.0.1, this requirement does not apply and there is no specific service-level requirement for the migration.

- If you are migrating from WebSphere Process Server for z/OS version 6.0.2.x to version 6.1 and your configuration uses XA connections, you must have version 6.0.2.9 or later before migrating.
- If you plan on running a IIOP client that is earlier than version 6.1 and it will be interacting with a version 6.1 server on the same LPAR, then the daemon procedure library for the version 6.1 daemon will need to include the earlier release's SBBOLD2 and SBBOLPA libraries in its STEPLIB.
- Migration support requires that both the source and target WebSphere Process Server for z/OS systems are on the same LPAR.

Therefore, you cannot migrate an existing configuration to a different z/OS LPAR. You also cannot migrate to or from a non-z/OS operating system using the WebSphere Process Server version 6.1 migration utilities.

- Migrating cells that span Sysplex environments or operating systems should not present any unique migration issues. You migrate at the node level, and you use the tools provided based on the platform of the node that you are migrating.

Note: For information on setting up a mixed-platform cell, see the *WebSphere for z/OS -- Heterogeneous Cells* white paper that is available at <http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100644>.

- You must generate the migration jobs using the Customization Dialog or z/OS Migration Management Tool and then submit them according to the generated instructions.
- Before you migrate to JDK 5 (introduced in WebSphere Application Server version 6.1, and therefore in WebSphere Process Server) from JDK 1.4 (used in prior versions of WebSphere Process Server), review your applications for necessary changes based on the Sun Microsystems Java™ specification. See API and specification migration.
- When migrating a cell with multiple nodes, the applications must remain at lowest JDK level until all nodes are migrated.
- WebSphere Process Server version 6.1 can be installed in an environment where it coexists with prior levels of WebSphere Process Server.

For information about coexistence, see *Coexisting with other WebSphere product installations*

Consider the following items while planning to enable coexistence:

- Update prerequisites to the levels required by WebSphere Process Server version 6.1.
Prior levels of WebSphere Process Server will continue to run at the higher prerequisite levels.
- In a mixed-mode cluster, if you start a version 6.1 cluster member, you cannot start any stopped version 6.0.x members remaining in that cluster. All currently running cluster members will continue to run and process messages normally until stopped. Once a version 6.0.x cluster member stops in a mixed-mode cell, you must migrate that member to version version 6.1 before starting the member again.
- Set up WebSphere Process Server for z/OS version 6.1 to eliminate potential LPA conflicts with a prior version 6.0.x installation.

Version 6.0.x, and 6.1 require the placement of some code into LPA (SBBOLPA). In addition, additional product code (SBBOLOAD) should be placed into LPA for performance reasons. Because of naming conflicts, however, more than one version of product code can not be in LPA at the same time.

- Review the ports that have been defined to ensure that the WebSphere Process Server version 6.1 installation does not conflict.

In particular, note that the default daemon port definition for both versions is the same when installing to coexist with WebSphere Process Server version 6.0.x.

See *Coexisting with other WebSphere product installations*.

- Consider the following if you are planning to have any mixed-release cells:
 - You can upgrade a portion of the nodes in a cell to WebSphere Process Server version 6.1 while leaving others at the older release level. This means that, for a period of time, you might be administering servers that are at the current release and servers that are running the newer release in the same cell.
 - A WebSphere Process Server version 6.1 deployment cell can contain mixed releases of 6.0.x and 6.1 nodes, but there is no mixed-node management support for version 6.0.1.x.

At version 6.1, the deployment manager can manage both version 6.1 nodes and version 6.0.2 nodes. The version 6.1 deployment manager cannot, however, manage version 6.0.1 nodes. If you have any version 6.0.1.x nodes in the deployment manager cell, you must perform one of the following tasks:

- Upgrade all version 6.0.1.x nodes to at least version 6.0.2.
- Immediately migrate these nodes to version 6.1.

- During migration, version 6.1 cluster information is distributed throughout the cell. Version 6.0.x nodes that are not at WebSphere Application Server version 6.0.2.11 or later fail to read this information and the cluster function might fail. Therefore, upgrade all version 6.0.x nodes that will be contained in or interoperating with a version 6.1 cell to version 6.0.2.11 or later before migrating your deployment managers to version 6.1.
- WebSphere Process Server version 6.1 migration converts HTTP transports to channel-framework Web container transport chains.
- Include maintenance considerations when developing your configuration file system strategy.

If you configure your network deployment environment using the default value for the product file system path in the Customization Dialog, it will result in all the nodes pointing directly at the mount point of the product file system. This makes rolling maintenance in a nondisruptive manner almost impossible. If a cell is configured in this way, applying service to the product file system affects all the nodes at the same time; and if multiple cells are configured in this way, applying service to the product file system affects all the cells at the same time.

You might want to specify what is referred to as an "intermediate symbolic link" between each node's configuration file system and the actual mount point of the product file system. This strategy is described in the *WebSphere Application Server for z/OS V5 - Planning for Test, Production and Maintenance* white paper.

See the *Washington Systems Center Sample WebSphere for z/OS ND Configuration* white paper for more information about this issue and its relationship to applying maintenance. See the *WebSphere for z/OS: Updating an Existing Configuration HFS to Use Intermediate Symbolic Links* instructions for information on obtaining and using a utility that would allow you to update an existing configuration file system to use intermediate symbolic links.

- The migration tools create a migration backup directory containing a backup copy of the configuration from the previous version. The space available for this

directory should be at least the size of the previous version's configuration directory and applications plus the size of the batch-job output from the migration.

Normally, the batch-job output from the migration is very small unless you enable trace. The trace output size varies depending on the parts of migration for which you have enabled trace. The largest trace producer is the WBIPostUpgrade phase of migration. You can typically see trace output of around 30 MB for this phase.

- Migrating to version 6.1 might increase the Java virtual machine (JVM) heap sizes for your servers on z/OS. The following heap sizes are required in version 6.1:
 - Control region minheap must be at least 48 MB, and maxheap must be no less than 256 MB.
 - Servant minheap must be at least 256 MB, and maxheap must be no less than 512 MB.

If your JVM heap sizes in the version that you are migrating are lower than these required values, the migration process increases them to the version 6.1 minimum values.

- WebSphere Process Server version 6.1 does not support the DB2® for zOS Local JDBC Provider (RRS).

If you use the DB2 for zOS Local JDBC Provider (RRS) in the version 6.0.x configuration that you are going to migrate, you must change your configuration to use the DB2 Universal JDBC Driver Provider before or immediately after you migrate to version 6.1. The version 6.1 migration tools do not migrate the provider for you.

If you use the DB2 for zOS Local JDBC Provider (RRS) in the version to be migrated and do not change your configuration to use the DB2 Universal JDBC Driver Provider before migrating to version 6.1, the following events will occur:

- When running the migration tools, you will receive the following message:
MIGR0442W: Not migrating DB2 for zOS Local JDBC Provider (RRS) jdbc provider. Manually create a DB2 Universal Driver provider as a replacement. See DB2 documentation for further details.
- After migration, DB2 access will be broken and you will receive the following runtime message:
DSRA8213W: JDBC provider, DB2 for zOS Local JDBC Provider (RRS), is no longer supported by WebSphere Application Server. Applications should use DB2 Universal JDBC Driver Provider Type 2.

If you determine that you must change your configuration to use the DB2 Universal JDBC Driver Provider, you can do so by performing one of the following tasks:

- Before migrating to version 6.1, change your version 6.0.x configuration to use the DB2 Universal JDBC Driver Provider.

If you do this, the version 6.1 migration tools will handle the migration to the DB2 Universal JDBC Driver Provider and there will be no postmigration activity required.

Perform™ one of the following actions:

- Manually change your configuration to use the DB2 Universal JDBC Driver Provider.

Search the information center for your version 6.0.2 product to find information on configuring a DB2 Universal JDBC Driver Provider for WebSphere Process Server for z/OS.

- Use the JDBC Migration Utility for DB2 on z/OS to migrate from the DB2 for zOS Local JDBC Provider (RRS) to the DB2 Universal JDBC Driver Provider.

This tool is a Jython script that migrates DB2 for zOS Local JDBC Providers (RRS) to DB2 Universal JDBC Driver Providers one node at a time. A white paper that accompanies the tool explains how to install and configure the DB2 Universal JDBC Driver before running the tool to migrate your configuration. The tool and white paper are available from the product support site at <http://www.ibm.com/support/docview.wss?uid=swg27007826>.

- After migrating to version 6.1, perform one of the following actions:
 - Manually change your configuration to use the DB2 Universal JDBC Driver Provider.
 - Use the JDBC Migration Utility for DB2 on z/OS to migrate from the DB2 for zOS Local JDBC Provider (RRS) to the DB2 Universal JDBC Driver Provider.

This tool is a Jython script that migrates DB2 for zOS Local JDBC Providers (RRS) to DB2 Universal JDBC Driver Providers one node at a time. A white paper that accompanies the tool explains how to install and configure the DB2 Universal JDBC Driver before running the tool to migrate your configuration. The tool and white paper are available from the product support site at <http://www.ibm.com/support/docview.wss?uid=swg27007826>.

- After you migrate a server to WebSphere Process Server for z/OS version 6.1, the administrative and user applications continue to be defined under the virtual host default_host as they were in the previous release. However, a migrated deployment manager is defined under the virtual host admin_host that was introduced in version 6.1.
- Before you migrate a Cloudscape® database, ensure that any servers hosting applications that are using the Cloudscape database are shut down. Otherwise, the Cloudscape migration will fail.
- After you use the migration tools to migrate to WebSphere Process Server version 6.1, you might need to do some things that are not done automatically by the migration tools.
 - Examine any Lightweight Third Party Authentication (LTPA) security settings that you might have used in version 6.0.x, and make sure that version 6.1 security is set appropriately.
 - If necessary, create new System Authorization Facility (SAF) profiles before your migrated servers are started on WebSphere Process Server for z/OS version 6.1.

Beginning with version 6.1, certain security facilities are controlled using SAF profiles.

- In version 6.1 and later, the Enabling Trusted Applications setting is controlled with a SAF security profile instead of an internal WebSphere variable as in previous releases.

The Enabling Trusted Applications option, which permits the WebSphere Process Server for z/OS runtime to perform certain privileged operations on behalf of application code, is required for all WebSphere Process Server for z/OS servers that use the LocalOS registry or SAF authorization.

- In version 6.1 and later, the Sync to OS Thread feature (which allows an application to access resources using an operating system identity other

than the server identity) is controlled with a SAF security profile and the `com.ibm.websphere.security.SyncToOSThread` variable.

This allows both the administrator and the system security administrator to determine whether or not the feature is used. This implementation also allows restrictions on which identities the application can assume.

If you migrate from a previous version of WebSphere Process Server and need these features, you should create the required SAF profiles. If these profiles are not present and properly set up, a cell using the LocalOS user registry or SAF authorization will fail when brought up on version 6.1.

If you use Resource Access Control Facility (RACF®) for your security system, use the following instructions. If you use another SAF-compliant security system, contact the security system vendor for appropriate information.

- Check your MVS™ system log or use the administrative console to determine whether or not Enable Trusted Applications is enabled for your server.

Look for `control_region_security_enable_trusted_applications` in the startup log; if it is set to 1, Enabled Trusted Applications is enabled. If it is enabled, create the following SAF profile and grant READ access to the application server control region user ID:

```
BBO.TRUSTEDAPPS.cell_shortname.cluster_transition_name
```

Use the following RACF commands to accomplish this:

```
RDEFINE FACILITY
  BBO.TRUSTEDAPPS.cell_shortname.cluster_transition_name
  UACC(NONE)
PERMIT FACILITY
  BBO.TRUSTEDAPPS.cell_shortname.cluster_transition_name
  ID(controller_userid) ACCESS(READ)
SETROPTS RACLIST(FACILITY) REFRESH
```

The `cluster_name` SAF facility profile is replaced by the cluster transition name for unclustered servers. If you want all servers in the cell to have Enabling Trusted Applications enabled, replace the cluster name with a wild card (*).

For more information, see System Authorization Facility classes and profiles

- Check your MVS system log or use the administrative console to determine whether or not Sync to OS Thread Allowed is enabled for your server.

If it is enabled, create the following SAF profile and grant either READ or CONTROL access to the application server control region user ID:

```
BBO.SYNCID.cell_shortname.cluster_transition_name
```

The following example contains RACF commands that you might use to accomplish this:

```
RDEFINE FACILITY
  BBO.SYNCID.cell_shortname.cluster_transition_name
  UACC(NONE)
PERMIT FACILITY
  BBO.SYNCID.cell_shortname.cluster_transition_name
  ID(controller_userid) ACCESS(CONTROL)
SETROPTS RACLIST(FACILITY) REFRESH
```

The cluster name is replaced by the cluster transition name for unclustered servers. If you want all servers in the cell to have Sync to OS Thread Allowed enabled, replace the cluster name with a wild card (*).

Notes:

- Granting the control region READ access to the application server control region user ID restricts the user IDs to which the thread identity can be changed based on SAF SURROGAT profiles.

If the controller user ID has READ access to the BBO.SYNC profile and the `com.ibm.websphere.security.SyncToOSThread` variable is set to true, an application might request Sync to the OS Thread. The application might assume the identity of either the caller or a role-related user ID to access resources as long as the new identity has READ access to the `BBO.SYNC.servant_user_ID` SAF SURROGAT profile.

- Granting the control region CONTROL access to the application server control region user ID allows the thread identity to be switched to any user ID that requests Sync to OS Thread.

If the controller user ID has CONTROL access to the BBO.SYNC profile and the `com.ibm.websphere.security.SyncToOSThread` variable is set to true, then an application might request Sync to OS Thread. The application might assume the identity of either the caller or any role-related user ID to access resources. SURROGAT profiles are not checked.

For more information, see [Application Synch to OS Thread Allowed](#).

- If you use SAF EJBROLE profiles for role-based authorization, create EJBROLE profiles for the two administrative roles—the deployer and `adminsecuritymanager` roles—that were introduced in version 6.1.
- Review your Java virtual machine settings to verify that you are using the recommended default values as described in [Java virtual machine settings](#).
- If you did not do so before migrating, back up any databases that support WebSphere Process Server components so that if you need to roll back the migration, you can also roll back the database.

Related concepts

“Migration considerations for Business Process Choreographer” on page 30
If your servers run Business Process Choreographer, you should be aware of some restrictions and additional tasks you might need to perform.

Related tasks

“Rolling back your environment” on page 25

After migrating to a WebSphere Process Server version 6.1 environment, you can roll back to a version 6.0.x environment. This returns the configuration to the state that it was in before migration. After rolling back the environment, you can restart the migration process.

Related reference

 [WBIPreUpgrade command](#)

Use the `WBIPreUpgrade` command for WebSphere Process Server to save the configuration of a previously installed version of WebSphere Process Server into a migration-specific backup directory.

 [WBIPostUpgrade command](#)

Use the `WBIPostUpgrade` command for WebSphere Process Server to retrieve the saved configuration that was created by the `WBIPreUpgrade` command from the `backupDirectory` that you specified. The `WBIPostUpgrade` command for WebSphere Process Server reads the configuration from this directory to

migrate to the more recent version of WebSphere Process Server and adds all migrated applications into the *profile_root/installedApps* directory for the new installation.

Related information

 Coexisting with other WebSphere product installations

An installation of WebSphere Process Server, version 6.1 can coexist on the same system with installations of any version of WebSphere Process Server or WebSphere Enterprise Service Bus, and with certain versions of selected WebSphere products.

Configuring transport chains

HTTP transport channel settings

Transport chains

API and specification migration

Creating clusters

Creating application servers

Lightweight Third Party Authentication

Core group migration considerations

Java virtual machine settings

How data is handled during migration from earlier versions

The WebSphere Process Server version-to-version migration tools will handle different sets of data (enterprise application data, configuration data, and system application data) in different ways.

Configuration data migration

The version-to-version migration scripts will automatically apply the configuration settings from the previous profile to the new profile created during the migration process. In cases in which the new profile has already been configured and values in the old profile and new profile do not match, the values will be handled as follows:

- The installation directory name that has already been configured in the new profile will be retained in the new profile.
- Any values from the old profile (other than the installation directory name) will replace non-matching values in the new profile.

Application migration

Your user applications (any applications not provided with the WebSphere Process Server product) are binary-compatible for the migration scenarios supported. (Refer to “Overview of migrating” on page 1 for supported migration scenarios.) All user applications will be automatically migrated to the new server. You should not have to modify any part of the application to have it run on the newer version of WebSphere Process Server.

Note: For version 6.0.1 WebSphere Adapters, some additional steps might be required for compatibility. For more information on this or any other exception, see the WebSphere Process Server technotes at the WebSphere Process Server technotes web site.

Except for sample applications, applications that are provided as part of the WebSphere Process Server product are migrated to the latest version of those applications. These are handled as follows:

- For all system applications—applications that reside in the *install_root* /systemApps directory, the newer version is installed.
- For all support applications—applications provided with WebSphere Process Server, such as the Business Rules Manager and Business Process Choreographer applications, older versions are updated to the latest version.

Related concepts

“Overview of migrating” on page 1

Migrate from earlier versions of WebSphere Process Server and WebSphere Enterprise Service Bus.

Configuration mapping during product-configuration migration

Various configurations are mapped during product-configuration migration.

Many migration scenarios are possible. The migration tools map objects and attributes existing in the version from which you are migrating to the corresponding objects and attributes in the newer version’s environment.

Command-line parameters

The migration tools convert appropriate command-line parameters to Java virtual machine (JVM) settings in the server process definition. Most settings are mapped directly. Some settings are not migrated because their roles in the WebSphere Process Server version 6.1 configuration do not exist, have different meanings, or have different scopes.

For information on how to change the process-definition settings, see Process definition settings in the WebSphere Application Server information center. For information on how to change the Java virtual machine settings, see Java virtual machine settings in the WebSphere Application Server information center.

Policy file

WebSphere Process Server version 6.1 migrates all the policy files that are installed with version 6.0.x policy files with the following characteristics:

- Any comments located in the version 6.1 policy file will be preserved. Any comments contained in the version 6.0.x policy file will not be included in the version 6.1.
- Migration will not attempt to merge permissions or grants; it is strictly an add-type migration. If the permission or grant is not located in the version 6.1 file, the migration will bring it over.
- Security is a critical component; thus, the migration makes any additions at the end of the original .policy file right after the comment MIGR03721: Migrated grant permissions follow. This is done to help administrators verify any policy file changes that the migration has made.

Properties and lib/app directories

Migration copies files from prior version directories into the WebSphere Process Server version 6.1 configuration.

Property files

WebSphere migrates all the WebSphere Process Server version 6.1 property files that are installed with version 6.0.x by merging settings into the version 6.1 property files.

Migration does not overlay property files.

Resource adapter archives (RARs) referenced by J2C resources

RARs that are referenced by J2C resources are migrated if those RARs are in the old WebSphere Process Server installation. In this case, the RARs are copied over to the corresponding location in the new WebSphere Process Server installation. Relational Resource Adapter RARs will not be migrated.

Migrating cluster-level resources:

Cluster-level resources are configured in resourcexxx.xml files under the cluster directories. For example:

```
<resources.j2c:J2CResourceAdapter xmi:id="J2CResourceAdapter_1112808424172"
  name="ims" archivePath="{WAS_INSTALL_ROOT}\installedConnectors\x2.rar">
  ...
</resources.j2c:J2CResourceAdapter>
```

If you have a cluster-level resource, this resource must be in the same location on each cluster member (node). Using the above example, therefore, each cluster member must have the RAR file installed at location `{WAS_INSTALL_ROOT}\installedConnectors\x2.rar`. `{WAS_INSTALL_ROOT}` is resolved on each cluster member to get the exact location.

In the migration of a deployment manager, the tools migrate the cluster files on the deployment manager, including the resourcexxx.xml files.

In the migration of a managed node, the tools process each J2C adapter. Files such as RAR files are migrated as follows from version 6.0.x to version 6.1.

Migration from version 6.0.x to version 6.1 copies files such as RAR files from `WAS_INSTALL_ROOT` to `WAS_INSTALL_ROOT` and from `USER_INSTALL_ROOT` to `USER_INSTALL_ROOT`.

If you have a RAR file in the `WAS_INSTALL_ROOT` for version 6.0.x, for example, the migration tools do not automatically copy the file from `WAS_INSTALL_ROOT` to `USER_INSTALL_ROOT`. This maintains the integrity of the cluster-level J2C resources. If you hardcoded a path to a RAR file (`archivePath="C:/WAS/installedConnectors/x2.rar"` for example) in version 6.0.x, however, the version 6.1 migration tools cannot change the `archivePath` attribute to reflect this because that would break all of the other cluster members that have not been migrated.

Samples

During the migration of the deployment manager, no WebSphere Process Server samples for federated nodes are migrated. Equivalent version 6.1 samples are available for all version 6.1 samples.

Security

Java 2 security is enabled by default when you enable security in WebSphere Process Server version 6.1. Java 2 security requires you to grant security permissions explicitly.

There are several techniques that you can use to define different levels of Java 2 security in version 6.1. One is to create a `was.policy` file as part of the application to enable all security permissions. The migration tools call the `wsadmin` command to add an existing `was.policy` file in the version 6.1 `properties` directory to enterprise applications as they are being migrated.

When migrating to WebSphere Process Server version 6.1, your choice of whether or not to migrate to support script compatibility results in one of two different outcomes.

- If you choose to migrate to support script compatibility, your security configuration is brought over to version 6.1 without any changes. This is the default.
- If you choose not to migrate to support script compatibility, the security configuration is converted to the default configuration for WebSphere Process Server version 6.1. The default version 6.1 security configuration acts almost the same as in the previous versions, but there are some changes.

For example, existing keyfiles and trustfiles are moved out of the `SSLConfig` repertoire and new keystore and truststore objects are created.

All SSL configuration repertoires of the System Secure Sockets Layer (SSSL) type, except those that belong to the `daemon`, are converted to the Java Secure Socket Extension (JSSE) type.

In order to retain the same security settings, you need to migrate the WebSphere Application Server security settings that might have been set for version 6.0.x. For more information on migrating your security configurations to version 6.1, see *Migrating, coexisting, and interoperating - Security considerations* in the WebSphere Application Server information center.

Stdin, stdout, stderr, passivation, and working directories

In WebSphere Process Server for z/OS, outputs for `stdin`, `stdout`, and `stderr` are directed to `SYSOUT` by default. If they are redirected to the configuration directory of a previous version, you might need to change this in the Version 6.1 JCL.

The migration tools attempt to migrate existing passivation and working directories. Otherwise, appropriate version 6.1 defaults are used.

For more information on passivation directories, see EJB container settings. For more information on working directories, see Process definition settings.

If WebSphere Process Server for z/OS user IDs have home directories in the configuration directory of a previous version, you should update them before migration to reside in another location.

In a coexistence scenario, using common directories between versions can create problems.

Transport ports

The migration tools migrate all ports. The tools log a port-conflict warning if a port is already defined in the configuration. You must resolve any port conflicts before you can run servers at the same time.

If the `-portBlock` parameter is specified in the `WBIPostUpgrade` part of the process, a new value is assigned to each transport that is migrated.

You must manually add virtual host alias entries for each port. For more information, see [Configuring virtual hosts](#).

Web modules

The specification level of the Java 2 Platform, Enterprise Edition (J2EE) implemented in WebSphere Process Server version 6.0.x required behavior changes in the Web container for setting the content type. If a default servlet writer does not set the content type, not only does the Web container no longer default to it but the Web container returns the call as "null." This situation might cause some browsers to display resulting Web container tags incorrectly. To prevent this problem from occurring, migration sets the `autoResponseEncoding` IBM® extension to "true" for Web modules as it migrates enterprise applications.

Related concepts

["Premigration considerations"](#) on page 2

Before you begin the process of migrating to a new version of WebSphere Process Server, you should be aware of these considerations.

["Migrating WebSphere applications"](#)

You should not have to modify any existing WebSphere Process Server applications to migrate them. However, more information about migrating different types of WebSphere applications is available in the WebSphere Application Server Information Center.

Related tasks

["Migrating a managed node"](#) on page 20

Migrate a version 6.0.x managed node to a version 6.1 managed node.

Related reference

 [WBIPostUpgrade command](#)

Use the `WBIPostUpgrade` command for WebSphere Process Server to retrieve the saved configuration that was created by the `WBIPreUpgrade` command from the `backupDirectory` that you specified. The `WBIPostUpgrade` command for WebSphere Process Server reads the configuration from this directory to migrate to the more recent version of WebSphere Process Server and adds all migrated applications into the `profile_root/installedApps` directory for the new installation.

Related information

[Process definition settings](#)

[Java virtual machine settings](#)

[Migrating, coexisting, and interoperating - Security considerations](#)

[EJB container settings](#)

[Transport chains](#)

[Configuring virtual hosts](#)

[Task overview: Using enterprise beans in applications](#)

Migrating WebSphere applications

You should not have to modify any existing WebSphere Process Server applications to migrate them. However, more information about migrating different types of WebSphere applications is available in the WebSphere Application Server Information Center.

Your applications—that is, any applications not provided with the WebSphere Process Server product—are binary-compatible for the migration scenarios

supported. (Refer to “Overview of migrating” on page 1 for supported migration scenarios.) You should not have to modify any part of the application to have it run on the newer version of WebSphere Process Server.

Note: For version 6.0.1 WebSphere Adapters, some additional steps might be required for compatibility. For more information on this or any other exception, see the WebSphere Process Server technotes at the WebSphere Process Server technotes web site.

For more information about migrating certain types of WebSphere applications, see Migrating WebSphere Applications and its subsidiary topics in the WebSphere Application Server Information Center. Because WebSphere Process Server is based upon WebSphere Application Server, the same information applies.

For information about migrating WebSphere Adapters, refer to the documentation for your adapter in the WebSphere Integration Developer documentation in the IBM WebSphere Business Process Management Version 6.1 information center .

Note that applications developed on a more recent version of WebSphere Process Server will not run on older versions. For details about run-time compatibility, see Development and deployment version levels.

Related concepts

“Overview of migrating” on page 1

Migrate from earlier versions of WebSphere Process Server and WebSphere Enterprise Service Bus.



Development and deployment version levels

Your decision about what version levels of WebSphere Process Server you need in your environment will depend on the version levels with which your applications were developed. Generally applications deployed in a previous version of WebSphere Process Server will run on the next available version of WebSphere Process Server.

Migrating stand-alone servers

Migrate a version 6.0.x stand-alone server to a version 6.1 stand-alone server.

Before you begin

Before you start:

- Read Techdoc WP100771: Migrating to WebSphere Application Server for z/OS Version 6.1.
- Have a version 6.0.x WebSphere Process Server server on z/OS.
- Install and configure a version 6.1 WebSphere Process Server server of the same type on z/OS. This server must have been augmented to use the same database as the version 6.0.x server uses.

Procedure

1. Stop the version 6.0.x server. See Starting a server from the MVS consoleStarting a server from the MVS console.
2. Back up the WebSphere Process Server database. If necessary, you can then recover the version 6.0.x system later.
3. Generate the migration jobs from the WebSphere Application Server Customization ISPF panels:

- a. In a TSO session, enter the following command:

```
ex 'high_level_qualifier.sbboclib(bbowstrt)' 'appl(bb61) lang(enus)'
```

Where *high_level_qualifier* is the high-level qualifier of the WebSphere Application Server installation libraries.

- b. Select **4 - Migrate a Node**, then select **1 Migrate a stand-alone application server node**. The WebSphere Application Server migration jobs are generated in two PDS data sets that you created when you worked through the WebSphere Application Server Customization ISPF panels. For example:

```
ZWPS.WAS.V602.V602SVR.MIG.CNTL  
ZWPS.WAS.V602.V602SVR.MIG.DATA
```

Where *V602SVR* is the name of the version 6.0.2 WebSphere Process Server server to be migrated.

There are detailed descriptions in the member *ZWPS.WAS.V602.V602SVR.MIG.CNTL(BBOMBINS)* of each job generated.

4. Customize the generated jobs to pick up the migration shell script. For a stand-alone server, only the following three jobs are required:
 - BBOWMG1B
 - BBOWMG2B
 - BBOWMG3B
 - a. In the installed WebSphere Process Server JCL PDS(*ZWPS.*.*.SBPZJCL*), locate the three sample WebSphere Process Server migration jobs that correspond to the following three jobs:
 - BPZWMG1B
 - BPZWMG2B
 - BPZWMG3B

These jobs invoke the WebSphere Process Server script *wbimigr2.sh*, which is very similar to the WebSphere Application Server script *bbomigr2.sh*. The *wbimigr2.sh* script invokes the migration utilities *WBIPreUpgrade.sh* and *WBIPostUpgrade.sh*.

- b. Edit the jobs so that they make use of the parameters generated by the WebSphere Application Server Customization panels, and are now present in the *BBOWMGxB* job.

BPZWMG1B and *BPZWMG2B* only need to be run if you had XA Connectors installed in the version 6.0.x server. *BPZWMG3B* does the actual migration.

5. Migrate the server:
 - a. Run the following jobs from PDS that were generated in the previous the previous step. For example, *ZWPS.WAS.V602.V602SVR.MIG.CNTL*.
 - b. If XA connectors were installed on the version 6.0.x server, run the *BPZWMG1B* and *BPZWMG2B* jobs.
 - c. Run the *BPZWMG3B* job.
6. Verify the migration. The migration process produces numerous diagnostic log files that need to be checked, including the following files:
 - All the log files in */WebSphere/V6R1/AppServer/profiles/default/logs/*
 - All the log files in the */tmp/migration/nnnnnn* directory that you specified in the migration job.

Most of these files are generated as ASCII files, so you must convert them to EBCDIC if you want to view them from TSO. If the tools that you use to view,

edit, and run the scripts require the scripts to be in EBCDIC format, use the `iconv` command to convert the file to EBCDIC. For example:

```
iconv -t IBM-1047 -f ISO8859-1 WASPreUpgradeSummary.log >
WASPreUpgradeSummary_EBCDIC.log
```

7. Upgrade the WebSphere Process Server databases. If any of the WebSphere Process Server databases need upgrading, SQL scripts are generated in the WebSphere `servername/DeploymentManager/dbscripts` directory of the version 6.1 server (where `servername` is the name of the new version 6.1 server). These SQL scripts are generated in database-specific directories. For example,
`/WebSphere/V6R1/AppServer/dbscripts/CommonDB/DB2zOSV8/upgradeSchema602.sql`
`/WebSphere/V6R1/AppServer/dbscripts/ProcessChoreographer/DB2zOSV8/upgradeTablespaces602.sql`
`/WebSphere/V6R1/AppServer/dbscripts/ProcessChoreographer/DB2zOSV8/upgradeSchema602.sql`
 - a. Copy the scripts to your working directory.
 - b. Assign the appropriate permissions to the copies of the files; for example:
`chmod 755 upgradeSchema602.sql`
 - c. Edit the values in the file to suit your needs. Remember to convert them from ASCII to EBCDIC as necessary.
 - d. Run the customized scripts using the tool of your choice. For example, `DBUtility.sh` or `SPUFI`.
8. If you are using the Business Process Choreographer Observer, the migration process also generates a SQL script for the Observer product. You must run these SQL files to enable Observer on the version 6.1 server. Do not edit this SQL script. The SQL script is generated in the configuration file system in the following location: `/WebSphere/V6R1/AppServer/profiles/default/dbscripts/ProcessChoreographer/DB2zOSV8/dbname/dbschema/createSchema_Observer.sql`, where `dbname` is the name of the database; `dbschema` is the name of the database schema.
9. Replace the started task JCL members in `USER.PROCLIB` by running the `BBOMBPCP` job from `ZWPS.WAS.V602.V602SVR.MIG.CNTL` to replace the version 6.0.x started task members with the new version 6.1 versions.
10. Start the server.
11. Check for any startup errors in the `SYSLOG` output files in the server's address spaces.

Results

The stand-alone server is migrated to version 6.1.

Related concepts

“Migration considerations for Business Process Choreographer” on page 30
If your servers run Business Process Choreographer, you should be aware of some restrictions and additional tasks you might need to perform.

“Premigration considerations” on page 2

Before you begin the process of migrating to a new version of WebSphere Process Server, you should be aware of these considerations.

Related tasks

“Verifying migration” on page 23

Verify that your migration was successful by checking the log files and checking operation with the administrative console.

“Troubleshooting version-to-version migration” on page 32

Review this page for troubleshooting tips if you encounter problems while migrating from an older version of WebSphere Process Server.

“Postmigration configuration checking” on page 24

After migration, you should check some configuration settings. You might need to change them, or further configure the version 6.1 server.

Related reference

 [WBIPostUpgrade command](#)

Use the WBIPostUpgrade command for WebSphere Process Server to retrieve the saved configuration that was created by the WBIPreUpgrade command from the *backupDirectory* that you specified. The WBIPostUpgrade command for WebSphere Process Server reads the configuration from this directory to migrate to the more recent version of WebSphere Process Server and adds all migrated applications into the *profile_root/installedApps* directory for the new installation.

 [WBIPreUpgrade command](#)

Use the WBIPreUpgrade command for WebSphere Process Server to save the configuration of a previously installed version of WebSphere Process Server into a migration-specific backup directory.

Related information

[backupConfig command](#)

[stopServer command](#)

Migrating a network deployment environment

You can migrate a network deployment environment from one version to another, without the need to reconfigure any components in the environment.

About this task

The following steps describe how to migrate a WebSphere Process Server network deployment configuration that has one deployment manager and two managed nodes.

Migrating a version 6.0.x node to a version 6.1 node

You can migrate a WebSphere Process Server node that belongs to a cell to version 6.1 without removing the node from the cell.

Migrate the deployment manager first, before migrating any base nodes in the cell.

Use the same cell name when migrating from version 6.0.x to version 6.1 . If you use a different cell name, federated nodes cannot successfully migrate to the version 6.1 cell.

Migrating a base WebSphere Process Server version 6.1 node that is within a cell also migrates the node agent to version 6.1. A cell can have some version 6.1 and other nodes that are at version 6.0.x.

Note: If, before migration, Business Rules Manager is running on any of the deployment targets (servers or clusters), you should review “Migrating Business Rules Manager in a network deployment environment” on page 22 before starting to migrate.

Procedure

1. “Migrating a deployment manager” on page 18
2. “Migrating a managed node” on page 20

Migrating a deployment manager

Migrate a version 6.0.x deployment manager to a version 6.1 deployment manager.

Before you begin

Do the following before you start migrating a deployment manager:

- Read Techdoc WP100771: Migrating to WebSphere Application Server for z/OS Version 6.1.
- Install and configure a version 6.1 WebSphere Process Server network deployment configuration of the same type on z/OS. The version 6.1 configuration must have been augmented to use the same database as the version 6.0.x configuration uses.

Procedure

1. On the version 6.0.x system, generate the migration jobs from the WebSphere Application Server Customization ISPF panels:
 - a. In a TSO session, enter the following command:

```
ex 'high_level_qualifier.sbboclib(bbowstrt)' 'appl(bb61) lang(enus)'
```

Where *high_level_qualifier* is the high-level qualifier of the WebSphere Application Server installation libraries.
 - b. Select **4 - Migrate a Node**, then select **2 - Migrate a deployment manager**. The WebSphere Application Server migration jobs are generated in two PDS data sets that you created when you worked through the WebSphere Application Server Customization ISPF panels.
2. Customize the generated migration jobs to pick up the user supplied parameters. For a deployment manager, only the job BPZWMG3D needs to be customized.
 - a. In the installed WebSphere Process Server JCL PDS(ZWPS.**.SBPZJCL), locate the sample WebSphere Process Server migration job BPZWMG3D. This job invokes the WebSphere Process Server script `wbimigr2.sh`, which is very similar to the WebSphere Application Server script `bbomigr2.sh`. The `wbimigr2.sh` script invokes the migration utilities `WBIPreUpgrade.sh` and `WBIPostUpgrade.sh`.
 - b. Edit the job BPZWMG3D to make use of the parameters generated by the WebSphere Application Server Customization panels, which are now present in the BPZWMG3D job.
3. Stop the version 6.0.x deployment manager. See Starting a server from the MVS consoleStarting a server from the MVS console.
4. Back up the WebSphere Process Server database. If necessary, you can then recover the version 6.0.x system later.
5. Submit the BPZWMG3D job that you edited.
6. Examine the output in `/tmp/migrate/XXXXXX/BPZWMG3D.out` and make a note of the name of the command that is in the message that contains the following text:

```
[wsadmin] You must manually update the Process Choreographer cluster 'ClusterT4' when half of the nodes have been migrated
```

You will need to run the command after migrating half of the nodes, as described in the message.

7. Upgrade the WebSphere Process Server databases. If any of the WebSphere Process Server databases need upgrading, SQL scripts are generated in the

WebSphere *servername*/DeploymentManager/dbscripts directory of the version 6.1 server (where *servername* is the name of the new version 6.1 server). These SQL scripts are generated in database-specific directories. For example,

```
/WebSphere/V6R1/DeploymentManager/dbscripts/CommonDB/DB2zOSV8/  
upgradeSchema602.sql  
/WebSphere/V6R1/DeploymentManager/dbscripts/ProcessChoreographer/DB2zOSV8/  
upgradeTablespaces602.sql  
/WebSphere/V6R1/DeploymentManager/dbscripts/ProcessChoreographer/DB2zOSV8/  
upgradeSchema602.sql
```

- a. Copy the scripts to your working directory.
 - b. Assign the appropriate permissions to the copies of the files; for example:

```
chmod 755 upgradeSchema602.sql
```
 - c. Edit the values in the file to suit your needs. Remember to convert them from ASCII to EBCDIC as necessary.
 - d. Run the customized scripts using the tool of your choice. For example, DBUtility.sh or SPUFI.
8. Update the started task JCL members in USER.PROCLIB by running the BBOMDCP job from the jcl library that was generated. The job replaces the version 6.0.x started task members with the new version 6.1 members.
 9. Start the deployment manager. See Starting a server from the MVS console Starting a server from the MVS console.

Results

The deployment manager is migrated to version 6.1.

What to do next

Next, migrate each of the managed nodes in the cell. See “Migrating a managed node” on page 20.

Related concepts

“Migration considerations for Business Process Choreographer” on page 30
If your servers run Business Process Choreographer, you should be aware of some restrictions and additional tasks you might need to perform.

“Premigration considerations” on page 2

Before you begin the process of migrating to a new version of WebSphere Process Server, you should be aware of these considerations.

Related tasks

“Migrating a managed node” on page 20

Migrate a version 6.0.x managed node to a version 6.1 managed node.

“Verifying migration” on page 23

Verify that your migration was successful by checking the log files and checking operation with the administrative console.

“Troubleshooting version-to-version migration” on page 32

Review this page for troubleshooting tips if you encounter problems while migrating from an older version of WebSphere Process Server.

“Postmigration configuration checking” on page 24

After migration, you should check some configuration settings. You might need to change them, or further configure the version 6.1 server.

“Rolling back a deployment cell” on page 25

You can use the **restoreConfig** and **wsadmin** commands to roll back a migrated WebSphere Process Server version 6.1 deployment cell to version 6.0.x. This

returns the configuration to the state that it was in before migration. After rolling back the deployment cell, you can restart the migration process.

Related reference

 [WBIPostUpgrade command](#)

Use the WBIPostUpgrade command for WebSphere Process Server to retrieve the saved configuration that was created by the WBIPreUpgrade command from the *backupDirectory* that you specified. The WBIPostUpgrade command for WebSphere Process Server reads the configuration from this directory to migrate to the more recent version of WebSphere Process Server and adds all migrated applications into the *profile_root/installedApps* directory for the new installation.

 [WBIPreUpgrade command](#)

Use the WBIPreUpgrade command for WebSphere Process Server to save the configuration of a previously installed version of WebSphere Process Server into a migration-specific backup directory.

Related information

[backupConfig command](#)

[stopServer command](#)

 [Coexisting with other WebSphere product installations](#)

An installation of WebSphere Process Server, version 6.1 can coexist on the same system with installations of any version of WebSphere Process Server or WebSphere Enterprise Service Bus, and with certain versions of selected WebSphere products.

Migrating a managed node

Migrate a version 6.0.x managed node to a version 6.1 managed node.

Before you begin

Before migrating any managed nodes in the cell, you must first migrate the deployment manager. See [Migrating a deployment manager](#).

Procedure

1. On the version 6.0.x system, generate the migration jobs from the WebSphere Application Server Customization ISPF panels:
 - a. In a TSO session, enter the following command:

```
ex 'high_level_qualifier.sbboclib(bbowstrt)' 'appl(bb61) lang(enus)'
```

Where *high_level_qualifier* is the high-level qualifier of the WebSphere Application Server installation libraries.
 - b. Select **4 - Migrate a Node**, then select **3 - Migrate a federated node**. The WebSphere Application Server migration jobs are generated in two PDS data sets that you created when you worked through the WebSphere Application Server Customization ISPF panels.
2. Customize the generated migration jobs to pick up the user supplied parameters. For a managed node, customize the following jobs: BPZWMG1F, BPZWMG2F, and BPZWMG3F.
 - a. In the installed WebSphere Process Server JCL PDS(ZWPS.**.SBPZJCL), locate the corresponding sample WebSphere Process Server migration jobs BPZWMG1F; BPZWMG2F and BPZWMG3F. These jobs invoke the

WebSphere Process Server script `wbimigr2.sh`, which is very similar to the WebSphere Application Server script `bbomigr2.sh`. The `wbimigr2.sh` script invokes the migration utilities `WBIPreUpgrade.sh` and `WBIPostUpgrade.sh`.

- b. Edit the BPZWMGxF jobs to make use of the parameters generated by the WebSphere Application Server Customization panels, which are now present in the BBOWMGxF jobs.
3. Stop the version 6.0.x managed node server but ensure that the deployment manager is running. See [Stopping a server](#) for more information.
4. Back up the WebSphere Process Server database. If necessary, you can then recover the version 6.0.x system later.
5. Submit the BPZWMGxF jobs that you edited. BPZWMG1F and BPZWMG2F only need to be run if you had XA Connectors installed in the version 6.0.x server. BPZWMG3F does the actual migration.
6. If you have migrated half the managed nodes, ensure that all the servers are stopped and the deployment manager is running, then run the `wsadmin.sh` command on the deployment manager. See [“Migrating a deployment manager”](#) on page 18. If any of the nodes are still running, the command fails. For example:

```
wsadmin.sh -f ProcessChoreographer/config/bpeupgrade.jacl -cluster ClusterT4 -migrationFrom 6.0.2.1
```
7. Update the started task JCL members in USER.PROCLIB by running the BBOMMCP job from the jcl library that was generated. The job replaces the version 6.0.x started task members with the new version 6.1 members.
8. Start the version 6.1 managed node server. See [Starting a server from the MVS console](#) for more information.

Related concepts

[“Premigration considerations”](#) on page 2

Before you begin the process of migrating to a new version of WebSphere Process Server, you should be aware of these considerations.

Related tasks

[“Migrating a deployment manager”](#) on page 18

Migrate a version 6.0.x deployment manager to a version 6.1 deployment manager.

[“Rolling back a managed node”](#) on page 28

You can use the **restoreConfig** and **wsadmin** commands to roll back a migrated WebSphere Process Server version 6.1 managed node to the state that it was in before migration. For each managed node that you want to roll back, you must roll back the managed node itself and the corresponding changes made to the master repository located on the deployment manager.

[“Postmigration configuration checking”](#) on page 24

After migration, you should check some configuration settings. You might need to change them, or further configure the version 6.1 server.

Related reference

[WBIPostUpgrade command](#)

Use the `WBIPostUpgrade` command for WebSphere Process Server to retrieve the saved configuration that was created by the `WBIPreUpgrade` command from the `backupDirectory` that you specified. The `WBIPostUpgrade` command for WebSphere Process Server reads the configuration from this directory to migrate to the more recent version of WebSphere Process Server and adds all migrated applications into the `profile_root/installedApps` directory for the new installation.

WBIPreUpgrade command

Use the WBIPreUpgrade command for WebSphere Process Server to save the configuration of a previously installed version of WebSphere Process Server into a migration-specific backup directory.

Related information

backupConfig command

stopServer command

Migrating Business Rules Manager in a network deployment environment

As part of the migration of a network deployment environment, the Business Rules Manager application that has been deployed to servers or clusters in the cell is not migrated automatically until the migration of the last node in the cell. As a result, you might need to manually migrate Business Rules Manager if it is running on a server or cluster that is not the last node to be migrated.

Business Rules Manager from previous versions of WebSphere Process Server is not compatible with WebSphere Process Server 6.1. In addition, as part of the migration of the deployment manager or nodes, any instances of the Business Rules Manager application that have been deployed to servers or clusters in the cell are not migrated automatically until the migration of the last node in the cell. The migration is not automatically performed earlier to allow for migration at a time that is most convenient to the environment. Environments in which the deployment targets (servers or clusters) where the Business Rules Manager is deployed that are not being migrated immediately can benefit by continuing to manage the business rules in their cells through the existing (previous version) Business Rules Manager.

However, an error occurs if the node that contains the deployment target of the Business Rules Manager is migrated to version 6.1 and the Business Rules Manager running on that deployment target has not yet been upgraded to version 6.1. To prevent this error from occurring, you can manually migrate Business Rules manager before performing the normal migration of the deployment target.

If the cell is running in mixed mode, where the cell contains WebSphere Process Server 6.1 nodes and nodes of earlier versions, management of the business rules through the Business Rules Manager of an earlier version can continue until an application with business rules that are created with WebSphere Integration Developer 6.1 is installed into the cell. Business rules from this version of WebSphere Integration Developer contain capabilities that are not supported by previous versions of the Business Rules Manager, and modifications to these rules can cause loss to the business rules and cause the business rules to not function correctly.

Note: For migration of a stand-alone profile from a previous version of WebSphere Process Server to version 6.1, the Business Rules Manager application is automatically migrated as part of the migration process. If the Business Rules Manager is installed in the profile, it is migrated. No additional configuration is required.

Migrating Business Rules Manager on the deployment target

To migrate the Business Rules Manager on a deployment target, use the wsadmin command.

Procedure

Run the wsadmin command as follows, depending on the type of deployment target:

If the deployment target is...	Run this command:
a server	<code>install_root/bin/wsadmin -f installBRManager.jacl -s server_name -n node_name</code>
a cluster	<code>install_root/bin/wsadmin -f installBRManager.jacl -cl cluster_name</code>
multiple targets (if the current Business Rules Manager is mapped to more than one target)	<code>install_root/bin/wsadmin -f installBRManager.jacl -m "{{target1} {target2} ... {targetn}}"</code> where each {targetn} is either { -s server_name -n node_name } or { -cl cluster_name }

Migrate the rest of the WebSphere Process Server software on the deployment target by following the appropriate steps described in “Migrating a network deployment environment” on page 17.

Verifying migration

Verify that your migration was successful by checking the log files and checking operation with the administrative console.

Before you begin

Make sure the server that has been migrated has been started.

Procedure

1. Check the migration log files for the WBIPostUpgrade command and the WBIProfileUpgrade.ant script.
 - a. Check the file `backupDirectory/logs/WBIPostUpgradetimestamp.log` for either of the following messages: (`backupDirectory` is the directory in which migrated data was first stored and later retrieved from during the migration process, as specified in the migration wizard or the WBIPreUpgrade or WBIPostUpgrade commands.)
 - MIGR0259I: The migration has successfully completed.
 - MIGR0271W: Migration completed successfully, with one or more warnings.
 - b. Check the file `backupDirectory/logs/WBIProfileUpgrade.anttimestamp.log` for the message BUILD SUCCESSFUL.

Both of these log files must indicate success, as described by these messages, for you to consider the migration successful.

2. Check the log files in the logs directory. For example, check the logs for a stand-alone server in the `/WebSphere/V6R1/AppServer/profiles/default/logs` directory.
3. Check operation with the administrative console.
 - a. Open the administrative console (Integrated Solutions Console).
 - b. Select **Applications > Enterprise Applications** from the navigation panel.

- c. In the right-hand panel, verify that all of the applications listed have started, shown by the green "started" icon.
- d. From the navigation panel, select **Resources > JDBC > Business Integration Data Sources**.
- e. For each WebSphere Process Server data source listed on this panel, select the check box and then select **Test connection**.
- f. For each data source, you should receive a message similar to the following:
"The test connection operation for data source WPS_DataSource on server Dmgr1 at node Dmgr1Node1 was successful."

What to do next

If migration was successful, you can begin using the server. If the migration was not successful, refer to "Troubleshooting version-to-version migration" on page 32 for troubleshooting information.

Related tasks

"Rolling back your environment" on page 25

After migrating to a WebSphere Process Server version 6.1 environment, you can roll back to a version 6.0.x environment. This returns the configuration to the state that it was in before migration. After rolling back the environment, you can restart the migration process.

"Troubleshooting version-to-version migration" on page 32

Review this page for troubleshooting tips if you encounter problems while migrating from an older version of WebSphere Process Server.

Related information

 [Administering enterprise applications](#)

Use the console's Enterprise Application page (viewed by clicking **Applications > Enterprise Applications**) to view and administer enterprise applications installed on the server.

Postmigration configuration checking

After migration, you should check some configuration settings. You might need to change them, or further configure the version 6.1 server.

Before you begin

You should have migrated your server or cluster and verified that the migration was successful.

About this task

Perform the following checks, if applicable to your environment:

- Examine any Lightweight Third Party Authentication (LTPA) security settings that you might have used in version 6.0.x, and make sure that version 6.1 security is set appropriately.
- Check the WBIPostUpgrade.log file in the logs directory for details about any JSP objects that the migration tools did not migrate.

If version 6.1 does not support a level for which JSP objects are configured, the migration tools recognize the objects in the output and log them.

- Review your Java virtual machine settings to verify that you are using the recommended heap sizes. See Java virtual machine settings. The information at this link applies to WebSphere Process Server servers as well as WebSphere Application Server servers.

Rolling back your environment

After migrating to a WebSphere Process Server version 6.1 environment, you can roll back to a version 6.0.x environment. This returns the configuration to the state that it was in before migration. After rolling back the environment, you can restart the migration process.

About this task

Generally, migration does not modify anything in the configuration of the prior release; however, there are cases where minimal changes are made that are reversible—those of a deployment manager and its managed nodes.

The subtopics below provide further information for these cases.

Related concepts

“Premigration considerations” on page 2

Before you begin the process of migrating to a new version of WebSphere Process Server, you should be aware of these considerations.

Related tasks

“Verifying migration” on page 23

Verify that your migration was successful by checking the log files and checking operation with the administrative console.

Rolling back a deployment cell

You can use the **restoreConfig** and **wsadmin** commands to roll back a migrated WebSphere Process Server version 6.1 deployment cell to version 6.0.x. This returns the configuration to the state that it was in before migration. After rolling back the deployment cell, you can restart the migration process.

Before you begin

When migrating a version 6.0.x deployment cell, you must complete the following if you want to be able to roll it back to its previous state after migration:

1. Back up the databases that support WebSphere Process Server components.
2. Back up your existing configuration using the **backupConfig** command or your own preferred backup utility.
 - Run the **backupConfig** command or your own preferred utility to back up the version 6.0.x deployment manager configuration.

Important: Make sure that you note the exact name and location of this backed-up configuration.

See the **backupConfig** command in the WebSphere Application Server information center.

- Run the **backupConfig** command or your own preferred utility to back up the version 6.0.x managed node configurations.

Important: Make sure that you note the exact name and location of each of these backed-up configurations.

See the backupConfig command in the WebSphere Application Server information center.

3. Migrate the deployment cell.

Procedure

1. Stop all of the servers that are currently running in the WebSphere Process Server version 6.1 environment.
2. If you chose to disable the previous deployment manager when you migrated to the version 6.1 deployment manager, perform one of the following actions.
 - a. If you backed up your previous deployment manager configuration using the **backupConfig** command or your own preferred backup utility, run the **restoreConfig** command or your own preferred utility to restore the version 6.0.x configuration for the deployment manager.

Important: Make sure that you restore the same backed-up configuration that you created just before you migrated the deployment manager.

See the restoreConfig command in the WebSphere Application Server information center.

- b. If you did not back up your previous deployment manager configuration, use the **wsadmin** command to run the migrationDisablementReversal.jacl script from the Version 5.x or 6.0.x WAS_HOME/bin directory of the deployment manager.

Use the following parameters:

```
./wsadmin.sh -f migrationDisablementReversal.jacl -conntype NONE
```

Tip: If you have trouble running the migrationDisablementReversal.jacl script, try to manually perform the steps in the script.

- 1) Go to the following directory:

```
WAS_HOME/config/cells/cell_name/nodes/node_name
```

where *node_name* is the name of the deployment manager node that you want to roll back.

- 2) If you see a serverindex.xml_disabled file in in this directory, perform the following actions:
 - a) Delete or rename the serverindex.xml file.
 - b) Rename the serverindex.xml_disabled file to serverindex.xml.
3. Perform one of the following actions for each of the deployment cell's managed nodes that you need to roll back.
 - a. If you backed up your previous managed node configuration using the **backupConfig** command or your own preferred backup utility, run the **restoreConfig** command or your own preferred utility to restore the version 6.0.x configuration for the managed node.

Important: Make sure that you restore the same backed-up configuration that you created just before you migrated the managed node.

See the restoreConfig command in the WebSphere Application Server information center.

- b. If you did not back up your previous managed node configuration, use the **wsadmin** command to run the migrationDisablementReversal.jacl script from the version 6.0.x *profile_root*/bin directory of the managed node.

Tip: If you have trouble running the migrationDisablementReversal.jacl script, try to manually perform the steps in the script.

1) Go to the following directory:

```
profile_root/config/cells/cell_name/nodes/node_name
```

where *node_name* is the name of the managed node that you want to roll back.

2) If you see a serverindex.xml_disabled file in in this directory, perform the following actions:

a) Delete or rename the serverindex.xml file.

b) Rename the serverindex.xml_disabled file to serverindex.xml.

c. If you did not back up your previous managed node configuration, use the **wsadmin** command to run the migrationDisablementReversal.jacl script from the version 6.0.x *install_root/bin* directory of the managed node.

Use the following parameters:

```
./wsadmin.sh -f migrationDisablementReversal.jacl -conntype NONE
```

Tip: If you have trouble running the migrationDisablementReversal.jacl script, try to manually perform the steps in the script.

1) Go to the following directory:

```
install_root/config/cells/cell_name/nodes/node_name
```

where *node_name* is the name of the managed node that you want to roll back.

2) If you see a serverindex.xml_disabled file in in this directory, perform the following actions:

a) Delete or rename the serverindex.xml file.

b) Rename the serverindex.xml_disabled file to serverindex.xml.

4. Synchronize the managed nodes if they were ever running when the version 6.1 deployment manager was running.

See Synchronizing nodes with the wsadmin tool in the WebSphere Application Server information center.

5. If you chose to keep the installed applications in the same location as the prior release during migration to version 6.1 and any of the version 6.1 applications are not compatible with the prior release, install applications that are compatible.

6. Delete the version 6.1 profiles.

See Deleting a profile in the WebSphere Application Server Information Center.

7. Roll back your databases. (For any databases that support WebSphere Process Server components that were upgraded, either automatically with the migration tools or manually, restore the backups that you made before you started the migration process.)

8. Start the rolled-back deployment manager and its managed nodes in the version 6.0.x environment.

Results

The configuration should now be returned to the state that it was in before migration.

What to do next

You can now restart the migration process if you want to do so.

Related tasks

“Rolling back a managed node”

You can use the **restoreConfig** and **wsadmin** commands to roll back a migrated WebSphere Process Server version 6.1 managed node to the state that it was in before migration. For each managed node that you want to roll back, you must roll back the managed node itself and the corresponding changes made to the master repository located on the deployment manager.

“Migrating a deployment manager” on page 18

Migrate a version 6.0.x deployment manager to a version 6.1 deployment manager.

Related information

restoreConfig command

backupConfig command

Synchronizing nodes with the wsadmin tool

Rolling back a managed node

You can use the **restoreConfig** and **wsadmin** commands to roll back a migrated WebSphere Process Server version 6.1 managed node to the state that it was in before migration. For each managed node that you want to roll back, you must roll back the managed node itself and the corresponding changes made to the master repository located on the deployment manager.

Before you begin

When you migrate a version 6.0.x managed node, you must complete the following if you want to be able to roll it back to its previous state after migration:

1. Back up the databases that support WebSphere Process Server components.
2. Back up your existing configuration using the **backupConfig** command or your own preferred backup utility.
 - Run the **backupConfig** command or your own preferred utility to back up the version 6.0.x deployment manager configuration..

Important: Make sure that you note the exact name and location of this backed-up configuration.

See the backupConfig command in the WebSphere Application Server information center.

- Run the **backupConfig** command or your own preferred utility to back up the version 6.0.x managed node configuration.

Important: Make sure that you note the exact name and location of this backed-up configuration.

See the backupConfig command in the WebSphere Application Server information center.

3. Migrate the managed node.

If necessary, you can now roll back the managed node that you just migrated.

Important: If you do not have a backup copy of your version 6.1 deployment manager configuration as it was before you migrated the version 6.0.x managed

node that you want to roll back, you cannot use the procedure described in this article and you must roll back your whole cell as described in “Rolling back a deployment cell” on page 25.

About this task

You must perform all of the backup and rollback actions for each migrated managed node before you proceed to roll back another managed node.

Procedure

1. Roll back your databases. (For any databases that support WebSphere Process Server components that were upgraded, either automatically with the migration tools or manually, restore the backups that you made before you started the migration process.)
2. Stop all of the servers that are currently running in the version 6.1 environment.
3. Restore your previous configuration.
 - a. Run the **restoreConfig** command or your own preferred utility to restore the version 6.1 deployment manager configuration.

Important: Make sure that you restore the same backed-up configuration that you created just before you migrated the managed node.

See the `restoreConfig` command in the WebSphere Application Server Information Center.

- b. Perform one of the following actions to restore the version 6.0.x configuration for the managed node.
 - Run the **restoreConfig** command or your own preferred utility to restore the version 6.0.x configuration.
See `restoreConfig` command in the WebSphere Application Server Information Center.
 - Use the **wsadmin** command to run the `migrationDisablementReversal.jacl` script from the version 6.0.x `install_root/bin` directory of the managed node that you need to roll back from version 6.1.

Use the following parameters:

```
./wsadmin.sh -f migrationDisablementReversal.jacl -conntype NONE
```

Tip: If you have trouble running the `migrationDisablementReversal.jacl` script, try to manually perform the steps in the script.

- 1) Go to the following directory:

```
WAS_HOME/config/cells/cell_name/nodes/node_name
```

where `node_name` is the name of the managed node that you want to roll back.

- 2) If you see a `serverindex.xml_disabled` file in in this directory, perform the following actions:
 - a) Delete or rename the `serverindex.xml` file.
 - b) Rename the `serverindex.xml_disabled` file to `serverindex.xml`.
4. Start the version 6.1 deployment manager.
5. Synchronize the managed node.
See Synchronizing nodes with the `wsadmin` tool in the WebSphere Application Server information center.

6. If you chose to keep the installed applications in the same location as the prior release during migration to version 6.1 and any of the version 6.1 applications are not compatible with the prior release, install applications that are compatible.
7. Start the rolled-back managed node in the version 6.1 environment.

Results

The configuration should now be returned to the state that it was in before migration.

What to do next

You can now restart the migration process if you want to do so.

Related tasks

“Rolling back a deployment cell” on page 25

You can use the **restoreConfig** and **wsadmin** commands to roll back a migrated WebSphere Process Server version 6.1 deployment cell to version 6.0.x. This returns the configuration to the state that it was in before migration. After rolling back the deployment cell, you can restart the migration process.

“Migrating a managed node” on page 20

Migrate a version 6.0.x managed node to a version 6.1 managed node.

Related information

restoreConfig command

backupConfig command

Synchronizing nodes with the wsadmin tool

Migration considerations for Business Process Choreographer

If your servers run Business Process Choreographer, you should be aware of some restrictions and additional tasks you might need to perform.

Mixed cell restrictions

If, during the course of the migration process to WebSphere Process Server 6.1.2, your cell is running nodes at both versions 6.0.x and 6.1.x concurrently or running nodes at versions 6.1.0 and 6.1.2 concurrently, be aware of the following:

- When a deployment manager has been migrated to version 6.1.2, you no longer can install, update, or uninstall Business Process Choreographer applications (BPEL applications or human tasks) on nodes in the cell that are still at the version 6.0.x level or 6.1.0 level.
- When a deployment manager has been migrated to version 6.1.2, you no longer can configure Business Process Choreographer on nodes in the cell that are still at the version 6.0.x level or the version 6.1.0 level.
- If you have a version 6.1.2 cluster with Business Process Choreographer configured on it, you must not create new cluster members on 6.0.x nodes or on 6.1.0 nodes in the same cell.

Postmigration tasks

You might need to perform these tasks, if they apply to your environment, before using your WebSphere Process Server version 6.1 in production.

- If your WebSphere Process Server version 6.0.1 used the Business Process Choreographer Observer sample, remove the sample. See Removing the Business Process Choreographer Observer Sample Version 6.0.1 Removing the Business Process Choreographer Observer Sample Version 6.0.1. This sample is not migrated. Business Process Choreographer Observer for version 6.0.2 or later versions is not a sample.
- If you have written a client that uses Business Process Choreographer APIs without first authenticating the user, you should modify the client to perform a login before using the APIs. After migration, the J2EE roles BPEAPIUser and TaskAPIUser are set to the value Everyone, which maintains backward compatibility by maintaining the 6.0.x behavior of not requiring a login when application security is enabled. After you have fixed your client, you must change these roles to the value AllAuthenticated to prevent unauthenticated users accessing the APIs. For new installations these roles default to the value AllAuthenticated.

To do this:

1. Open the administrative console and select **Applications > Enterprise Applications**.
 2. In the right panel, select BPEContainer_ *name* , where *name* is either *nodeName_serverName* or *clusterName*, depending on whether you configured Business Process Choreographer on a server or on a cluster. (Select the name, not the check box to the left of the name.)
 3. In the right panel, under Detail Properties, select **Security role to user/group mapping**.
 4. Change the mapping for the J2EE BPEAPIUser role from "Everyone" to "All authenticated".
 5. Select **OK**.
 6. Repeat these steps for the TaskAPIUser role of the TaskContainer_ *name* enterprise application.
 7. Save your changes, then restart the server or cluster on which you configured Business Process Choreographer.
- If you have applied any changes to the default XSL transformation files (EverybodyTransformation.xml, LDAPTransformation.xml, SystemTransformation.xml, and UserRegistryTransformation.xml) located in the *install_root/ProcessChoreographer/Staff* directory then you must re-apply your changes to the WebSphere Process Server version 6.1 versions of these files after migration. Custom XSL transformation files located in the *install_root/ProcessChoreographer/Staff* directory will be migrated automatically. Custom XSL transformation files located in other directories may have to be copied manually, depending on the exact value of the transformation file path specified in the version 6.0.x staff plug-in configuration (now known as the people directory configuration in WebSphere Process Server version 6.1).

Related concepts

"Premigration considerations" on page 2


Before you begin the process of migrating to a new version of WebSphere Process Server, you should be aware of these considerations.

Related tasks

"Troubleshooting version-to-version migration" on page 32

Review this page for troubleshooting tips if you encounter problems while migrating from an older version of WebSphere Process Server.

Related information

 Removing the Business Process Choreographer Observer Version 6.0.1 sample

 Administering enterprise applications

Use the console's Enterprise Application page (viewed by clicking **Applications** > **Enterprise Applications**) to view and administer enterprise applications installed on the server.

Troubleshooting version-to-version migration

Review this page for troubleshooting tips if you encounter problems while migrating from an older version of WebSphere Process Server.

- If you encounter a problem when you are migrating from a previous version of WebSphere Process Server to version 6.1, check your log files and other available information.
 1. Look for diagnostic information in the two files that were generated when you ran the migration job. The files were written to the JESOUT data set during the WROUT and WRERR steps; browse them from SDSF.
 2. Look in the following log files, which are in ASCII format (if you are viewing them on z/OS, you must convert them to EBCDIC first):
 - *migration_backup_directory*/base_backup/WASPreUpgrade.time_stamp.log
 - *migration_backup_directory*/base_backup/WBIPostUpgrade.time_stamp.log
 - *migration_backup_directory*/base_backup/WBIPreUpgrade.time_stamp.log
 - /WebSphere/V6R1/DeploymentManager/profiles/default/logs/WASPreUpgradeSummary.log
 - /WebSphere/V6R1/DeploymentManager/profiles/default/logs/WASPostUpgradeSummary.log
 - /WebSphere/V6R1/DeploymentManager/profiles/default/logs/WASPostUpgrade.time_stamp.log
 - /WebSphere/V6R1/AppServer/profiles/default/logs/WASPreUpgradeSummary.log
 - /WebSphere/V6R1/AppServer/profiles/default/logs/WASPostUpgradeSummary.log
 - /WebSphere/V6R1/AppServer/profiles/default/logs/WASPostUpgrade.time_stamp.log
 3. Look for the following messages in the logs:

MIGR0259I: The migration has successfully completed.

MIGR0271W: The migration completed with warnings.
 4. Open the Log and Trace Analyzer built into the Application Server Toolkit (AST) on the service log of the server that is hosting the resource that you are trying to access, and use it to browse error and warning messages.

See Debugging components in the Application Server Toolkit.
 5. With WebSphere Process Server, run the dumpNameSpace command and pipe, redirect, or "more" the output so that it can be easily viewed.

This command results in a display of all objects in WebSphere Process Server namespace, including the directory path and object name.
 6. If the object a client needs to access does not appear, use the administrative console to verify the following conditions.
 - The server hosting the target resource is started.

- The Web module or Enterprise JavaBean container hosting the target resource is running.
- The JNDI name of the target resource is properly specified.

If none of these steps solves the problem, see Troubleshooting and support for additional troubleshooting resources, including information about how to contact IBM Support.

- During the migration process, problems might occur during the WBIPreUpgrade or WBIPostUpgrade steps.
 - Problems can occur during the WBIPreUpgrade step.
 - A "Not found" or "No such file or directory" message is returned.
This problem can occur if the WBIPreUpgrade script is not in the correct version 6.1 bin directory; for example: /WebSphere/V6R1/DeploymentManager/profiles/default/bin. Verify that the WBIPreUpgrade script is in the correct directory so that the migration job can run the script.
 - The DB2 JDBC driver and DB2 JDBC driver (XA) cannot be found in the drop-down list of supported JDBC providers in the administrative console. The administrative console no longer displays deprecated JDBC provider names. The new JDBC provider names used in the administrative console are more descriptive and less confusing. The new providers will differ only by name from the deprecated ones.
The deprecated names will continue to exist in the jdbc-resource-provider-templates.xml file for migration reasons (for existing JACL scripts for example); however, you are encouraged to use the new JDBC provider names in your JACL scripts.
 - You receive the following message:
MIGR0108E: The specified WebSphere directory does not contain a WebSphere version that can be upgraded.
This can occur if an incorrect directory was used in the WBIPreUpgrade step of the migration job.
 - Problems can occur during the WBIPostUpgrade step.
 - A "Not found" or "No such file or directory" message is returned.
This problem can occur if the WBIPostUpgrade script is not in the correct version 6.1 bin directory; for example: /WebSphere/V6R1/DeploymentManager/profiles/default/bin. Verify that the WBIPostUpgrade script is in the correct directory so that the migration job can run the script.
 - When you migrate the federated nodes in a cell, you receive the following error messages:
MIGR0304I: The previous WebSphere environment is being restored.
com.ibm.websphere.management.exception.RepositoryException:
com.ibm.websphere.management.exception.ConnectorException: ADMC0009E:
The system failed to make the SOAP RPC call: invoke
MIGR0286E: The migration failed to complete.
A connection timeout occurs when the federated node tries to retrieve configuration updates from the deployment manager during the WBIPostUpgrade migration step for the federated node. Copying the entire configuration might take more than the connection timeout if the configuration that you are migrating to version 6.1 contains any of the following elements:
 - Many small applications
 - A few large applications
 - One very large application

If this occurs, modify the timeout value before running the migration job.

1. Change to the properties directory for the version 6.1 profile to which you are migrating your federated node; for example:
`/WebSphere/V6R1/AppServer/profiles/default/properties`
2. Open the `soap.client.props` file in that directory and find the value for the `com.ibm.SOAP.requestTimeout` property. This is the timeout value in seconds. The default value is 180 seconds.
3. Change the value of `com.ibm.SOAP.requestTimeout` to make it large enough to migrate your configuration. For example, the following entry would give you a timeout value of a half of an hour:
`com.ibm.SOAP.requestTimeout=1800`

Note: Select the smallest timeout value that will meet your needs. Be prepared to wait for at least three times the timeout that you select—once to download files to the backup directory, once to upload the migrated files to the deployment manager, and once to synchronize the deployment manager with the migrated node agent.

4. Go to the following location in the backup directory that was created by the `WBIPreUpgrade` step of the migration job:
`migration_backup_directory/profiles/default/properties`
5. Open the `soap.client.props` file in that directory and find the value for the `com.ibm.SOAP.requestTimeout` property.
6. Change the value of `com.ibm.SOAP.requestTimeout` to the same value that you used in the version 6.1 file.

- You receive the "Unable to copy document to temp file" error message. Here is an example:

```
MIGR0304I: The previous WebSphere environment is being restored.  
com.ibm.websphere.management.exception.DocumentIOException: Unable to copy  
document to temp file:  
cells/sunblade1Network/applications/LARGEApp.ear/LARGEApp.ear
```

Your file system might be full. If your file system is full, clear some space and rerun the `WBIPostUpgrade` command.

- You receive the following message:

```
MIGR0108E: The specified WebSphere directory does not contain a WebSphere  
version that can be upgraded.
```

The following possible reasons for this error exist:

- An incorrect directory might have been used when running the `WBIPreUpgrade` or `WBIPostUpgrade` steps
- The `WBIPreUpgrade` command was not run.

- You receive the following error message:

```
MIGR0253E: The backup directory migration_backup_directory does not exist.
```

The following possible reasons for this error exist:

- An incorrect backup directory might be specified.

For example, the directory might have been a subdirectory of the version 6.0.x tree that was deleted after the `WBIPreUpgrade` command was run and the older version of the product was uninstalled but before the `WBIPostUpgrade` command was run.

1. Determine whether or not the full directory structure specified in the error message exists.
2. If possible, rerun the `WBIPreUpgrade` command, specifying the correct full migration backup directory.

3. If the backup directory does not exist and the older version it came from is gone, rebuild the older version from a backup repository or XML configuration file.

4. Rerun the WBIPreUpgrade command.

- You decide that you need to run WBIPreUpgrade again after you have already run the WBIPostUpgrade command.

During the course of a deployment manager or a managed node migration, WBIPostUpgrade might disable the old environment. If after running WBIPostUpgrade you want to run WBIPreUpgrade again against the old installation, you must run the migrationDisablementReversal.jacl script located in the old *install_root*/bin directory. After running this JACL script, your version 6.0.x environment will be in a valid state again, allowing you to run WBIPreUpgrade to produce valid results.

For more information on scripting, see Getting started with scripting. Scripting, as described there, is available for WebSphere Process Server.

- A federated migration fails with message MIGR0405E.

The migration that has taken place on your deployment manager as part of your federated migration has failed. For a more detailed reason for why this error has occurred, open the folder *your_node_name_migration_temp* located on your deployment manager node under the ...DeploymentManagerProfile/temp directory. For example:

```
/websphere61/procserver/profiles/dm_profile/temp/nodeX_migration_temp
```

The logs and everything else involved in the migration for this node on the deployment manager node are located in this folder. This folder will also be required for IBM support related to this scenario.

- WebSphere Process Server version 6.1 applications are lost during migration.

If any of the version 6.1 applications fail to install during a federated migration, they will be lost during the synchronizing of the configurations. The reason that this happens is that one of the final steps of WBIPostUpgrade is to run a syncNode command. This has the result of downloading the configuration on the deployment manager node and overwriting the configuration on the federated node. If the applications fail to install, they will not be in the configuration located on the deployment manager node. To resolve this issue, manually install the applications after migration. If they are standard version 6.1 applications, they will be located in the *install_root*/installableApps directory.

To manually install an application that was lost during migration, use the wsadmin command to run the *install_application_name.jacl* script that the migration tools created in the backup directory.

See the Wsadmin tool.

- WebSphere Process Server version 6.1 applications fail to install.

Manually install the applications using the wsadmin command after WBIPostUpgrade has completed.

To manually install an application that failed to install during migration, use the wsadmin command to run the *install_application_name.jacl* script that the migration tools created in the backup directory.

See the Wsadmin tool, or see the WBIPostUpgrade command.

- If you select the option for the migration process to install the enterprise applications that exist in the version 6.0.x configuration into the new version 6.1

configuration, you might encounter some error messages during the application-installation phase of migration.

The applications that exist in the version 6.0.x configuration might have incorrect deployment information—usually, incorrect XML documents that were not validated sufficiently in previous WebSphere Process Server runtimes. The runtime now has an improved application-installation validation process and will fail to install these malformed EAR files. This results in a failure during the application-installation phase of WBIPostUpgrade and produces an "E:" error message. This is considered a "fatal" migration error.

If migration fails in this way during application installation, you can do one of the following:

- Fix the problems in the version 6.0.x applications, and then remigrate.
- Proceed with the migration and ignore these errors.

In this case, the migration process does not install the failing applications but does complete all of the other migration steps.

Later, you can fix the problems in the applications and then manually install them in the new version 6.1 configuration using the administrative console or an install script.

- After migrating to a version 6.1 cell that contains or interoperates with version 6.0.x nodes that are not at WebSphere Process Server version 6.0.1.3 or later, the cluster function might fail.

When starting these version 6.0.x servers, you might see the following problems:

- You might see a first failure data capture (FFDC) log that shows a `ClassNotFoundException` error message. This exception is thrown from the `RuleEtiquette.runRules` method and looks something like the following example:

```
Exception = java.lang.ClassNotFoundException
Source = com.ibm.ws.cluster.selection.SelectionAdvisor.<init>
probeid = 133
Stack Dump = java.lang.ClassNotFoundException: rule.local.server
at java.net.URLClassLoader.findClass(URLClassLoader.java(Compiled Code))
at com.ibm.ws.bootstrap.ExtClassLoader.findClass(ExtClassLoader.java:106)
at java.lang.ClassLoader.loadClass(ClassLoader.java(Compiled Code))
at java.lang.ClassLoader.loadClass(ClassLoader.java(Compiled Code))
at java.lang.Class.forName1(Native Method)
at java.lang.Class.forName(Class.java(Compiled Code))
at com.ibm.ws.cluster.selection.rule.RuleEtiquette.runRules(RuleEtiquette.java:154)
at com.ibm.ws.cluster.selection.SelectionAdvisor.handleNotification(SelectionAdvisor.java:153)
at com.ibm.websphere.cluster.topography.DescriptionFactory$Notifier.run(DescriptionFactory.java:257)
at com.ibm.ws.util.ThreadPool$Worker.run(ThreadPool.java:1462)
```

- You might see a `java.io.IOException` that looks something like the following example:

```
Exception = java.io.IOException
Source = com.ibm.ws.cluster.topography.DescriptionManagerA.update probeid = 362
Stack Dump = java.io.IOException
at com.ibm.ws.cluster.topography.ClusterDescriptionImpl.importFromStream(ClusterDescriptionImpl.java:916)
at com.ibm.ws.cluster.topography.DescriptionManagerA.update(DescriptionManagerA.java:360)
Caused by: java.io.EOFException
at java.io.DataInputStream.readFully(DataInputStream.java(Compiled Code))
at java.io.DataInputStream.readUTF(DataInputStream.java(Compiled Code))
at com.ibm.ws.cluster.topography.KeyRepositoryImpl.importFromStream(KeyRepositoryImpl.java:193)
```

During migration, version 6.1 cluster information is distributed throughout the cell. WebSphere Process Server version 6.0.x nodes that are not at version 6.0.1.3 or later fail to read this information.

To avoid this problem, upgrade all version 6.0.x nodes that will be contained in or interoperating with a version 6.1 cell to version 6.0.1.3 or later before migrating your deployment managers to version 6.1.

- After you migrate a managed node to version 6.1, the application server might not start.

When you try to start the application server, you might see errors similar to those in the following example:

```
[5/11/06 15:41:23:190 CDT] 0000000a SystemErr R
    com.ibm.ws.exception.RuntimeError:
com.ibm.ws.exception.RuntimeError: org.omg.CORBA.INTERNAL:
    CREATE_LISTENER_FAILED_4
vmcid: 0x49421000 minor code: 56 completed: No
[5/11/06 15:41:23:196 CDT] 0000000a SystemErr R at
com.ibm.ws.runtime.WsServerImpl.bootServerContainer(WsServerImpl.java:198)
[5/11/06 15:41:23:196 CDT] 0000000a SystemErr R at
com.ibm.ws.runtime.WsServerImpl.start(WsServerImpl.java:139)
[5/11/06 15:41:23:196 CDT] 0000000a SystemErr R at
com.ibm.ws.runtime.WsServerImpl.main(WsServerImpl.java:460)
[5/11/06 15:41:23:196 CDT] 0000000a SystemErr R at
com.ibm.ws.runtime.WsServer.main(WsServer.java:59)
[5/11/06 15:41:23:196 CDT] 0000000a SystemErr R at
sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
[5/11/06 15:41:23:196 CDT] 0000000a SystemErr R at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:64)
[5/11/06 15:41:23:197 CDT] 0000000a SystemErr R at
sun.reflect.DelegatingMethodAccessorImpl.invoke
    (DelegatingMethodAccessorImpl.java:43)
```

Change the port number at which the managed node's server is listening. If the deployment manager is listening at port 9101 for ORB_LISTENER_ADDRESS, for example, the server of the managed node should not be listening at port 9101 for its ORB_LISTENER_ADDRESS. To resolve the problem in this example, perform the following steps:

1. On the administrative console, click **Application servers** → *server_name* → **Ports** → ORB_LISTENER_ADDRESS.
 2. Change the ORB_LISTENER_ADDRESS port number to one that is not used.
- In a network deployment environment, if the error SRVE0026E: [Servlet Error]-[com/ibm/wbi/servers/brules/BusinessRuleManager]: java.lang.NoClassDefFoundError occurs when you access the Business Rules Manager after migrating, you must manually install the Business Rules Manager application on the deployment target before continuing with normal migration of that node. See Migrating Business Rules Manager in a network deployment environment for more information.
 - If synchronization fails when you migrate a managed node to version 6.1, the server might not start.

You might receive messages similar to the following when you migrate a managed node to version 6.1:

```
ADMU0016I: Synchronizing configuration between node and cell.
ADMU0111E: Program exiting with error:
    com.ibm.websphere.management.exception.AdminException: ADMU0005E:
    Error synchronizing repositories
ADMU0211I: Error details may be seen in the file:
    /opt/WebSphere/61AppServer/profiles/AppSrv02/logs/syncNode.log
MIGR0350W: Synchronization with the deployment manager using the SOAP protocol
    failed.
```

MIGR0307I: The restoration of the previous WebSphere Application Server environment is complete.

MIGR0271W: Migration completed successfully, with one or more warnings.

These messages indicate the following:

- Your deployment manager is at a version 6.1 configuration level.
- The managed node that you are trying to migrate is at a version 6.1 configuration level on the deployment manager's repository (including applications).
- The managed node itself is not quite complete given that you did not complete the syncNode operation.

Perform the following actions to resolve this issue:

1. Rerun the syncNode command on the node to synchronize it with the deployment manager.
See the syncNode command .
 2. Run the GenPluginCfg command.
See the GenPluginCfg command .
- If you encounter a problem when you are migrating from a previous version of WebSphere Process Server to version 6.1, check your log files and other available information.
 - If the migration job fails before the WBIPostUpgrade step, re-run the migration job.
 - If the migration job fails in the WBIPostUpgrade step, the configuration of the new 6.1 server is partially updated so re-create (or restore from backup) the new 6.1 server before re-running the migration job.
 - Problems occur with a managed (federated) node migration.

A federated node is the most complex node to migrate because it is essentially two migrations rolled into one. A federated node requires a migration of the node configuration information contained in the deployment manager's master repository as well as the configuration information contained in the federated node. Federated node migration requires an active connection to the deployment manager. If you have security enabled, it is essential that you follow the instructions that were generated when you created the migration jobs. The migration job must be submitted with a WebSphere Administrator's user ID that has been properly configured for obtaining secure connections.

version 6.x node agents might display as not synchronized or not available when you change the deployment manager node name in a mixed cell during migration to the version 6.1 deployment manager. version 6.x node agents maintain a link to the version 6.x deployment manager until they are restarted; therefore, they might fail to synchronize with the new deployment manager. The discovery problem, which prevents automatic synchronization, occurs because the node agent is not yet aware of the deployment manager name change that occurred during the migration. If you experience this problem, perform these steps on the node.

 1. Stop the node.
 2. Run the **syncNode** command.
 3. Restart the node.
 - Job fails during the application-installation phase of migration.

If you select the option for the migration process to install the enterprise applications that exist in the version 6.0.x configuration into the new version

6.1 configuration, you might encounter error messages during the application-installation phase of migration.

The applications that exist in the version 6.0.x configuration might have incorrect deployment information—usually, invalid XML documents that were not validated sufficiently in previous WebSphere Process Server runtimes. The runtime now has an improved application-installation validation process and will fail to install these malformed EAR files. This results in a failure during the application-installation phase of WBIPostUpgrade and produces an "E:" error message. This is considered a "fatal" migration error.

If migration fails in this way during application installation, you can do one of the following:

- Fix the problems in the version 6.0.x applications, and then remigrate.
- Proceed with the migration and ignore these errors.
 1. Restart the migration job in the FINISHUP step to allow the remaining migration functions to be performed.

Do this by adding the RESTART=FINISHUP parameter to the job card and resubmitting the job.
 2. Later, fix the problems in the applications and then manually install them in the new version 6.1 configuration using the administrative console or an install script.

- Out of space errors occur.

The migration logs are located in *temporary_directory_location/nnnnn*, where *temporary_directory_location* is the value that you specified when you created the migration jobs (where the default is */tmp/migrate*) and *nnnnn* is a unique number that was generated during the creation of your migration jobs.

Normally, the space requirements for the migration logs are small. If you enable tracing, however, the log files can be quite large. The best practice is to enable tracing only after problems have been found. If tracing is required, try to only enable tracing related to the step in the process that is being debugged. This will help to reduce the space requirements.

Tracing can be enabled when you create the migration jobs or by changing the variables in the migration JCL from disabled to enabled:

```
TraceState=enabled
profileTrace=disabled
preUpgradeTrace=disabled
postUpgradeTrace=enabled
```

During migration, a backup copy of your version 6.0.x configuration is made. This backup becomes the source of the information being migrated. The default backup location is */tmp/migrate/nnnnn*. This location can be changed when you create the migration jobs. Depending on the size of the node being migrated, this backup can be quite large. If your temporary space is inadequate, then you will need to relocate this backup.

- Batch job time is exceeded.

Each z/OS installation is different with respect to job classes and time limitations. Make sure you have specified appropriate job classes and timeout values on your job card.

- Failures occur during server startup after migration.

Review the instructions that were generated when you created the migration jobs. Verify that the JCL procedures have been copied over correctly to your PROCLIB, the RACF definitions have been created, the version 6.1 libraries have been authorized, and, if required, your STEPLIB statements to the version 6.1 libraries have been specified. Make sure that the daemon process

associated with your cell is at the appropriate level. The daemon process must be at the highest WebSphere Process Server for z/OS version level of all servers that it manages within the cell.

After migrating to a version 6.1 cell that contains or interoperates with version 6.0.x nodes that are not at version 6.0.1.3 or later, the cluster function might fail. When starting these version 6.0.x application servers, you might see the following problems:

- You might see a first failure data capture (FFDC) log that shows a `ClassNotFoundException` error message. This exception is thrown from the `RuleEtiquette.runRules` method and looks something like the following example:

```
Exception = java.lang.ClassNotFoundException
Source = com.ibm.ws.cluster.selection.SelectionAdvisor.<init>
probeid = 133
Stack Dump = java.lang.ClassNotFoundException: rule.local.server
at java.net.URLClassLoader.findClass(URLClassLoader.java(Compiled Code))
at com.ibm.ws.bootstrap.ExtClassLoader.findClass(ExtClassLoader.java:106)
at java.lang.ClassLoader.loadClass(ClassLoader.java(Compiled Code))
at java.lang.ClassLoader.loadClass(ClassLoader.java(Compiled Code))
at java.lang.Class.forName1(Native Method)
at java.lang.Class.forName(Class.java(Compiled Code))
at com.ibm.ws.cluster.selection.rule.RuleEtiquette.runRules
(RuleEtiquette.java:154)
at com.ibm.ws.cluster.selection.SelectionAdvisor.handleNotification
(SelectionAdvisor.java:153)
at com.ibm.websphere.cluster.topography.DescriptionFactory$Notifier.run
(DescriptionFactory.java:257)
at com.ibm.ws.util.ThreadPool$Worker.run(ThreadPool.java:1462)
```

- You might see a `java.io.IOException` that looks something like the following example:

```
Exception = java.io.IOException
Source = com.ibm.ws.cluster.topography.DescriptionManagerA.update probeid
= 362
Stack Dump = java.io.IOException
at com.ibm.ws.cluster.topography.ClusterDescriptionImpl.importFromStream
(ClusterDescriptionImpl.java:916)
at com.ibm.ws.cluster.topography.DescriptionManagerA.update
(DescriptionManagerA.java:360)
Caused by: java.io.EOFException
at java.io.DataInputStream.readFully(DataInputStream.java(Compiled Code))
at java.io.DataInputStream.readUTF(DataInputStream.java(Compiled Code))
at com.ibm.ws.cluster.topography.KeyRepositoryImpl.importFromStream
(KeyRepositoryImpl.java:193)
```

During migration, version 6.1 cluster information is distributed throughout the cell. Version 6.0.x nodes that are not at version 6.0.1.3 or later fail to read this information. To avoid this problem, upgrade all version 6.0.x nodes that will be contained in or interoperating with a version 6.1 cell to version 6.0.1.3 or later before migrating your deployment managers to version 6.1.

After migration, carefully review the job output and log files for errors.

Note: WebSphere Process Server provides an interactive problem control system (IPCS) verb `exit` to help you to format information from dumps of WebSphere Process Server processes. This verb `exit` was named `CBDATA`, which was an alias of the real module name, in version 6.0.x and earlier. In version 6.1, that alias was removed. In version 6.1 and later, therefore, you must use the real name of this verb `exit`, `BBORDATA`, instead of the alias.

If you migrate a node to version 6.1 then discover that you need to revert back to version 6.0.x, see [Rolling back your environment](#).

If none of these steps solves the problem, see [Troubleshooting and support](#) for additional troubleshooting resources, including information about how to contact IBM Support.

- In a network deployment environment, if the error SRVE0026E: [Servlet Error]-[com/ibm/wbi/servers/brules/BusinessRuleManager]: java.lang.NoClassDefFoundError occurs when you access the Business Rules Manager after migrating, you must manually install the Business Rules Manager application on the deployment target before continuing with normal migration of that node. See [Migrating Business Rules Manager in a network deployment environment](#) for more information.

What to do next

If you did not find your problem listed, contact IBM support.

Related concepts

“Migration considerations for Business Process Choreographer” on page 30
If your servers run Business Process Choreographer, you should be aware of some restrictions and additional tasks you might need to perform.

Related tasks

“Verifying migration” on page 23
Verify that your migration was successful by checking the log files and checking operation with the administrative console.

Related reference

WBIPreUpgrade command

Use the WBIPreUpgrade command for WebSphere Process Server to save the configuration of a previously installed version of WebSphere Process Server into a migration-specific backup directory.

WBIPostUpgrade command

Use the WBIPostUpgrade command for WebSphere Process Server to retrieve the saved configuration that was created by the WBIPreUpgrade command from the *backupDirectory* that you specified. The WBIPostUpgrade command for WebSphere Process Server reads the configuration from this directory to migrate to the more recent version of WebSphere Process Server and adds all migrated applications into the *profile_root/installedApps* directory for the new installation.

Related information

[Debugging components in the Application Server Toolkit](#)

[Wsadmin tool](#)

[syncNode command](#)

[GenPluginCfg command](#)

Troubleshooting and support

To help you understand, isolate, and resolve problems with your IBM software, the troubleshooting and support information contains instructions for using the problem-determination resources that are provided with your IBM products.

[Getting started with scripting](#)

Chapter 2. Migrating from previous WebSphere products

You can migrate applications and configuration data from certain IBM products that existed before WebSphere Process Server.

Migration from another product to WebSphere Process Server is supported from the following predecessor products:

- WebSphere Business Integration Server Foundation versions 5.1 and 5.1.1. For more information see “Migrating source artifacts from WebSphere Studio Application Developer Integration Edition.”
- WebSphere MQ Workflow version 3.6. For more information, see “Migrating from WebSphere MQ Workflow” on page 44.

Note: You can also migrate to WebSphere Process Server from certain versions of WebSphere Enterprise Service Bus and WebSphere Application Server, as well as from prior versions of WebSphere Process Server itself. For more information about migration from these products, see Chapter 1, “Migrating from previous versions of WebSphere Process Server and WebSphere Enterprise Service Bus,” on page 1.

For migration from a previous product to WebSphere Process Server (for example, WebSphere Business Integration Server Foundation for z/OS to WebSphere Process Server for z/OS), the migration steps include using migration tools to convert source artifacts to the new WebSphere Process Server version of the artifacts.

WebSphere Integration Developer contains migration tools that assist in migrating existing application source artifacts into WebSphere Process Server artifacts. These tools can be accessed through the **File > Import...** wizards of WebSphere Integration Developer. The migration tools designed to assist with migration from WebSphere InterChange Server can also be accessed through the command line of WebSphere Process Server.

You can also find articles that might help you with migration in the IBM developerWorks® “Technical Library” at <http://www.ibm.com/developerworks>.

Migrating source artifacts from WebSphere Studio Application Developer Integration Edition

To migrate source artifacts from WebSphere Studio Application Developer Integration Edition use the tools available in WebSphere Integration Developer.

About this task

Use the migration wizard or command line available from WebSphere Integration Developer to migrate WebSphere Application Server Developer Integration Edition service projects into projects in the active WebSphere Integration Developer workspace. Refer to the WebSphere Integration Developer information center for more information.

Related information

WebSphere Integration Developer information center

Migrating from WebSphere MQ Workflow

To migrate from WebSphere MQ Workflow, use the WebSphere Integration Developer migration wizard or a special utility to migrate from WebSphere MQ Workflow 3.6 to WebSphere Process Server.

About this task

For this version of WebSphere MQ Workflow...	Do this
WebSphere MQ Workflow 3.6	Use either the WebSphere Integration Developer migration wizard or the FDL2BPEL utility to migrate all WebSphere MQ Workflow artifacts into WebSphere Integration Developer deployable artifacts.
WebSphere MQ Workflow 3.5 or earlier	You must first migrate to WebSphere MQ Workflow version 3.6.

See the WebSphere Integration Developer information center for more information.

Related information

[WebSphere Integration Developer information center](#)

Chapter 3. Deprecated features

This section summarizes deprecated features in the product offerings comprising WebSphere Process Server version 6.0 and 6.1, and WebSphere Business Integration Server Foundation version 5.1. Deprecated features from other WebSphere Application Server version 5.1 and 6.x product offerings are described in the documentation for those products.

Deprecation list

This topic describes the deprecated features in the following versions and releases:

- “Deprecated features in WebSphere Process Server version 6.1.2”
- “Deprecated features in WebSphere Process server version 6.1” on page 46
- “Deprecated features in WebSphere Process Server version 6.0.2” on page 49
- “Deprecated features in WebSphere Process Server version 6.0.1” on page 51
- “Deprecated features in WebSphere Process Server version 6.0” on page 51
- “Deprecated features in WebSphere Business Integration Server Foundation version 5.1.1” on page 54
- “Deprecated features in WebSphere Business Integration Server Foundation version 5.1” on page 54

The following tables summarize what is deprecated, by version and release. Each table reflects the version and release where the deprecation took effect and lists what is being deprecated, such as features, APIs, scripting interfaces, tools, wizards, publicly exposed configuration data, naming identifiers, and constants. Where possible, a recommended migration action is provided.

Deprecated features in WebSphere Process Server version 6.1.2

WebSphere Process Server version 6.1.2 has no deprecated features.
--

Deprecated features in WebSphere Process server version 6.1

Container Manager Persistence over Anything (CMP/A)
<p>The CMP/A support included with WebSphere Process Server is deprecated. This includes the runtime support for applications which have been customized to use CMP/A, the cmpdeploy.bat/.sh command line tool, and the following public APIs:</p> <ul style="list-style-type: none">• com.ibm.websphere.rsadapter.WSProceduralPushDownHelper• com.ibm.websphere.rsadapter.WSPushDownHelper• com.ibm.websphere.rsadapter.WSPushDownHelperFactory• com.ibm.websphere.rsadapter.WSRelationalPushDownHelper <p>Recommended migration action:</p> <p>Convert CMP Entity Beans to use a relational data source, or have the CMP entity bean replaced by a different supported data persistence model.</p> <p>You can also use WebSphere Adapters to replace your existing CMP/A applications. The Adapter tools use a 'Create, Retrieve, Update, and Delete' architecture for creating service interfaces that is very similar to the architecture that CMP/A uses.</p>
JACL scripts (deprecated in WebSphere Application Server version 6.1)
<p>JACL script files are deprecated in WebSphere Process Server to maintain consistency with the deprecation of JACL scripts in WebSphere Application Server.</p> <p>Recommended migration action:</p> <p>Use the corresponding .bat/.sh files or wsadmin commands to perform the same functions. Note: The following Business Process Choreographer JACL scripts are not deprecated:</p> <ol style="list-style-type: none">1. <install_root>\ProcessChoreographer\admin\bpcTemplates.jacl2. <install_root>\ProcessChoreographer\config\bpeconfig.jacl3. <install_root>\ProcessChoreographer\config\bpeunconfig.jacl4. <install_root>\ProcessChoreographer\config\bpeupgrade.jacl5. <install_root>\ProcessChoreographer\config\clientconfig.jacl
IBM Web Services Client for C++
<p>The IBM Web Services Client for C++ is a standalone application with its own installer, but which is distributed on the WebSphere Process Server media. The product does not use or have a dependency on this software, however the IBM Message Service Client for C/C++ which is also distributed with the product does.</p> <p>Recommended migration action:</p> <p>Use one of the other freely available tools, such as gSOAP (http://www.cs.fsu.edu/~engelen/soap.html) which is an open source product distributed under the GPL license, which will provide the same functions.</p>
Business Process Choreographer
<p><u>Generic Business Process EJB API</u></p> <ul style="list-style-type: none">• The getAutoDelete() function from ProcessTemplateData is deprecated. <p>Recommended migration action:</p> <p>Use method getAutoDeletionMode() to query how auto deletion is handled for the corresponding process template.</p> <ul style="list-style-type: none">• The exception SpecificFaultReplyException is deprecated. <p>Recommended migration action:</p> <p>No action is required. This exception is only needed to handle WSIF messages, which are no longer supported.</p>

Container Manager Persistence over Anything (CMP/A)

Generic Business Process WebService API - XML schema types

Element `autoDelete` of the complex type `ProcessTemplateType` is deprecated

```
<xsd:element name="ProcessTemplate" type="tns:ProcessTemplateType"/>
<xsd:complexType name="ProcessTemplateType">
  <xsd:sequence>
    ...
    <xsd:element name="autoDelete" type="xsd:boolean" minOccurs="0"/>
  ...</xsd:sequence></xsd:complexType>
```

Recommended migration action:

Use element `autoDeletionMode` of type `ProcessTemplateType`

```
<xsd:element name="ProcessTemplate" type="tns:ProcessTemplateType"/>
<xsd:complexType name="ProcessTemplateType">
  <xsd:sequence>
    ...
    <xsd:element name="autoDeletionMode" type="xsd:string" minOccurs="0"/>
  ...</xsd:sequence></xsd:complexType>
```

Deprecation of Observer DB Cleanup Methods of the ProcessContainer MBean

The following methods are deprecated:

- `public String observerForceRemoveInstanceData(String dataSourceName, String state, String templateName, String validFrom, String completedBefore)`
- `public String observerRemoveDeletedInstancesData(String dataSourceName, String completedBefore)`
- `public String observerRemoveInstanceDataOfTemplate(String dataSourceName, String templateName, String validFrom)`

Recommended migration action:

Use the following new methods (with the same name and an additional Parameter 'dbSchemaName')

- `public String observerForceRemoveInstanceData(String dataSourceName, String dbSchemaName, String state, String templateName, String validFrom, String completedBefore)`
- `public String observerRemoveDeletedInstancesData(String dataSourceName, String dbSchemaName, String completedBefore)`
- `public String observerRemoveInstanceDataOfTemplate(String dataSourceName, String dbSchemaName, String templateName, String validFrom)`

LDAP staff resolution plug-in

The attribute evaluation specification for staff queries of the LDAP staff resolution plug-in is deprecated:

```
<slldap:attribute name="attribute name"
  objectclass="LDAP object class"
  usage="simple">
</slldap:attribute>
```

Recommended migration action:

Use the result object evaluation specification supporting multiple attributes per LDAP object. The attributes "objectclass" and "attribute" of the "user" query will be replaced by a full result object evaluation specification that supports multiple result attributes per person.

Container Manager Persistence over Anything (CMP/A)

Generic Human Task Manager EJB API

- The following fields from interface Task are deprecated:
 - STATE_FAILING
 - STATE_SKIPPED
 - STATE_STOPPED
 - STATE_TERMINATING
 - STATE_WAITING
 - STATE_PROCESSING_UNDO

Recommended migration action:

Use retrieve the staff activity associated with the inline human task for inline human tasks, and check the activity state using the `getExecutionState()` method on the `ActivityInstanceData` interface in the Generic Business Process EJB API.

- The field `KIND_WPC_STAFF_ACTIVITY` from interface Task is deprecated.

Recommended migration action:

Use the method `isInline()` on the Task interface to determine whether a human task is associated with a human task (staff) activity in a business process,

Deprecation of e-mail people assignment criteria

The e-mail receiver people assignment criteria (staff verbs) used for escalations with escalation action "e-mail" are deprecated, as they are not required anymore in version 6.1. This applies to the following people assignment criteria:

- Email Address for Department Members
- Email Address for Group Members
- Email Address for Group Members without Filtered Users
- Email Address for Group Search
- Email Address for Role Members
- Email Address for Users
- Email Address for Users by user ID

Recommended migration action:

E-mail addresses and the preferred language are resolved together with the user ID by the standard set of people assignment criteria with version 6.1. This deprecation information is thus especially important for those who write custom XSLT people assignment criteria mapping (staff verb) files. If you do not intend to deploy version 6.0.2 task definitions, you do not need to support the deprecated people assignment criteria. Note that with version 6.1, the people assignment criteria "User Records by user ID" has been introduced and has to be supported by custom XSLT files, since it resolves e-mail addresses as fallback.

You can eliminate the deprecated e-mail people assignment criteria in your existing human task definitions by initiating source artifact migration in WebSphere Integration Developer 6.1. To do this, import your version 6.0.2 task definition into WebSphere Integration Developer 6.1, make a minor change (like adding a blank to the task description and deleting it again), and then save it again.

Deprecation of MQ as JMS Provider for BPC-internal messaging (Configuration of business process container and human task container)

Configuring the business process container and human task container to use MQSeries® as the JMS provider is deprecated. The business process container and human task container use JMS for their internal messaging — specifically, to navigate long-running process instances.

Recommended migration action:

During configuration of the business process container and human task container, use the default JMS messaging provider.

Business Objects

Container Manager Persistence over Anything (CMP/A)
<p>The following Business Object methods are deprecated:</p> <ul style="list-style-type: none"> • com.ibm.websphere.bo.BOFactory.createByClass(java.lang.Class interfaceClass); • com.ibm.websphere.bo.BOType.getTypeByClass(java.lang.Class className); <p>Recommended migration action:</p> <p>These methods will raise "function not supported" exceptions if they are called in version 6.1</p>
Common Event Infrastructure
<p>Creation and editing of user-visible Common Base Events are deprecated.</p> <p>Recommended migration action:</p> <p>You can now use the tools to specify the Business Object data that is to be included in monitored emitted events</p>
zOS
<p>The requirement to bind a String object into JNDI at esb/messageLogger/qualifier is deprecated.</p> <p>Recommended migration action:</p> <p>The Message Logger primitives will now store message information within the CommonDB database. Where necessary, during the profile augmentation phase, a WebSphere variable called <code>ESB_MESSAGE_LOGGER_QUALIFIER</code> will now be created and its value set to that of the chosen CommonDB schema qualifier.</p>
WebSphere Enterprise Service Bus (WESB)
<p>The current method to identify an SSL repertoire to be used when WESB communicates with a secured WSRR instance has been deprecated.</p> <p>Recommended migration action:</p> <p>A new property has been added to WSRR definitions to allow the specification of such a repertoire.</p>

Deprecated features in WebSphere Process Server version 6.0.2

Human Task Manager
<p>The task context variable <code>%htm:task.clientDetailURL%</code> is no longer required, and thus has been deprecated.</p> <p>Recommended migration action:</p> <p>No action is required.</p>
<p>The standard e-mail implementation used for all escalation e-mails in TEL has been deprecated, and replaced by native support for defining e-mails in TEL.</p> <p>Recommended migration action:</p> <p>Use the customizable e-mail feature for escalations.</p>
<p>The following Task object methods that were deprecated in version 6.0 are no longer deprecated:</p> <ul style="list-style-type: none"> • getInputMessageTypeNames() • getOutputMessageTypeNames() <p>Recommended migration action:</p> <p>You can now use these methods.</p>
Business Process Choreographer

The method `getProcessAdministrators()` in the Generic Business Process EJB API interfaces `ActivityInstanceData`, `ProcessInstanceData`, and `ProcessTemplateData` are deprecated:

Recommended migration action:

Use these corresponding methods:

- `getProcessAdminTaskID()` in combination with method `getUsersInRole()` of the `HumanTaskManagerService` interface, as follows:
`htm.getUsersInRole(actInstData.getProcessAdminTaskID(), WorkItem.REASON_ADMINISTRATOR)`
- `getAdminTaskID()` in combination with method `getUsersInRole()` of the `HumanTaskManagerService` interface, as follows:
`htm.getUsersInRole(procInstData.getAdminTaskID(), WorkItem.REASON_ADMINISTRATOR)`
- `getAdminTaskTemplateID()` in combination with method `getUsersInRole()` of the `HumanTaskManagerService` interface, as follows:
`htm.getUsersInRole(procTemplData.getAdminTaskTemplateID(), WorkItem.REASON_ADMINISTRATOR)`

The following methods are deprecated for the `BusinessFlowManagerService` interface in Generic Business Process EJB API and the `HumanTaskManagerService` interface in the Generic Task EJB API:

- `query(String storedQueryName, Integer skipTuples)`
- `query(String storedQueryName, Integer skipTuples, Integer threshold)`

Recommended migration action:

Use these corresponding methods:

- `query(String storedQueryName, Integer skipTuples, List parameters)`
- `query(String storedQueryName, Integer skipTuples, Integer threshold, List parameters)`

SCA Admin Commands

The following commands (used with `wsadmin`) are deprecated:

- `configSCAForServer`
- `configSCAForCluster`

Recommended migration action:

Use these two commands in place of `configSCAForServer` for equivalent function. :

- `configSCAAsyncForServer`
- [Optional; use only if required] `configSCAJMSForServer`

Use these two commands in place of `configSCAForCluster` for equivalent function:

- `configSCAAsyncForCluster`
- [Optional; use only if required] `configSCAJMSForCluster`

The following JACL scripts are deprecated:

- `deleteAuditLog.jacl`
- `deleteInvalidProcessTemplate.jacl`
- `deleteInvalidTaskTemplate.jacl`
- `queryNumberOfFailedMessages.jacl`
- `replayFailedMessages.jacl`
- `cleanupUnusedStaffQueryInstances.jacl`
- `refreshStaffQuery.jacl`

Recommended migration action:

For each deprecated JACL scripts, a corresponding Jython script is now provided. Use the Jython scripts (*.py), which can be found in the `<install_root>/ProcessChoreographer/admin` directory.

Deprecated features in WebSphere Process Server version 6.0.1

WebSphere Process Server version 6.0.1 has no deprecated features.
--

Deprecated features in WebSphere Process Server version 6.0

Application programming model and container support features
--

The BRBeans component is deprecated, and is being replaced with business rules.

Recommended migration action:

You must manually remove all usages of BRBeans and move to business rules.

Some BPEL business process modeling constructs have been syntactically changed in version 6. Only the syntax is supported by WebSphere Integration Developer version 6.0. Migration is available for these constructs.
--

Recommended migration action:

Use the migration wizard provided by WebSphere Integration Developer to migrate WebSphere Business Integration Server Foundation version 5.1 service projects (including process definitions) to WebSphere Process Server version 6.0. After the migration wizard has finished, you must carry out some manual steps to complete the migration. For more information about migrating service projects, see the information center for WebSphere Integration Developer version 6.0.

In WebSphere Business Integration Server Foundation version 5.1, there is an option for the input of an undo service to implicitly provide a message that results from the merge of the input data of the compensable service overlaid by its output data . Given the enhanced compensation support provided by BPEL this functionality is deprecated.
--

Recommended migration action:

Use BPEL compensation for business processes.

Because of changes in the Business Flow Manager functionality In WebSphere Process Server version 6.0, the following methods are deprecated in the generic process API:

- The WorkList object has been renamed to StoredQuery; consequently, the following methods are deprecated on the BusinessFlowManager bean, and, if applicable, the methods you would use WebSphere Process Server version 6.0 are given:
 - newWorkList(String workListName, String selectClause, String whereClause, String orderByClause, Integer threshold, TimeZone timezone)
Replace with: createStoredQuery(String storedQueryName, String selectClause, String whereClause, String orderByClause, Integer threshold, TimeZone timezone)
 - getWorkListNames()
Replace with: getStoredQueryNames()
 - deleteWorkList(String workListName)
Replace with: deleteStoredQuery(String storedQueryName)
 - getWorkList(String workListName)
Replace with: getStoredQuery(String storedQueryName)
 - executeWorkList(String workListName)
Replace with: query(String storedQueryName, Integer skipTuples)
 - getWorkListActions()
not supported.
- The WorkListData object is deprecated.
Use StoredQueryData instead.
- The following methods of the ProcessTemplateData object are no longer supported:
 - getInputMessageTypeSystemName()
 - getOutputMessageTypeSystemName()
- The following methods of the ProcessInstanceData object are no longer supported:
 - getInputMessageTypeSystemName()
 - getOutputMessageTypeSystemName()
- The following methods of the ActivityInstanceData object are no longer supported:
 - getInputMessageTypeSystemName()
 - getOutputMessageTypeSystemName()
- The following methods of the ActivityServiceTemplateData object are no longer supported:
 - getInputMessageTypeSystemName()

Recommended migration action:

Use the replacement methods, if any, that are given.

Because of changes in the Human Task Manager functionality In WebSphere Process Server version 6.0, the following methods are deprecated in the generic process API

- The following methods are deprecated on the HumanTaskManager bean, and their replacements for use in WebSphere Process Server version 6.0 are given:
 - createMessage(TKIID tkiid, String messageTypeName)
Use the specific methods createInputMessage(TKIID tkiid), createOutputMessage(TKIID tkiid), createFaultMessage(TKIID tkiid) instead.
 - createMessage(String tkiid, String messageTypeName)
Use the specific methods createInputMessage(String tkiid), createOutputMessage(String tkiid), createFaultMessage(String tkiid) instead.
- For the Task object, the following methods are no longer supported:
 - getInputMessageTypeNames()
 - getOutputMessageTypeNames()

Recommended migration action:

Use the replacement methods, if any, that are given.

The following database views are deprecated:

- DESCRIPTION
- CUSTOM_PROPERTY

Recommended migration action:

Use the TASK_DESC view for the DESCRIPTION view and the TASK_CPROP view for the CUSTOM_PROPERTY view.

Programming Model of Java Code Snippets

- In WebSphere Business Integration Server Foundation version 5.1, access to BPEL variables within inline Java code snippets (activities and conditions) is provided through getter and setter methods. These methods are not supported. The WSIFMessage method that is used to represent BPEL variables in Java code snippets is also not supported.
- Methods <typeOfP> getCorrelationSet<cs> Property<p>() are not supported, because they do not consider correlation sets declared at the scope level; they can only be used to access correlation sets declared at the process level.
- The WebSphere Business Integration Server Foundation version 5.1 methods to access custom properties within Java snippet activities are not supported.
- The following getPartnerLink methods are not supported. Because they do not consider partner links declared on the scope level, they can only be used to access partner links declared at the process level.
 - EndpointReference getPartnerLink();
 - EndpointReference getPartnerLink (int role);
 - void setPartnerLink (EndpointReference epr);

Recommended migration action:

Use the migration wizard provided by WebSphere Integration Developer 6.0 to migrate WebSphere Business Integration Server Foundation version 5.1 service projects (including process definitions) to WebSphere Process Server version 6.0. After the migration wizard has finished, you must carry out some manual steps to complete the migration. For more information about migrating service projects, see the information center for WebSphere Integration Developer version 6.0.

Application services features

The Extended Messaging Service feature and all of the EMS/CMM APIs and SPIs are deprecated:

- com/ibm/websphere/ems/CMMCorrelator
- com/ibm/websphere/ems/CMMException
- com/ibm/websphere/ems/CMMReplyCorrelator
- com/ibm/websphere/ems/CMMRequest
- com/ibm/websphere/ems/CMMResponseCorrelator
- com/ibm/websphere/ems/ConfigurationException
- com/ibm/websphere/ems/FormatException
- com/ibm/websphere/ems/IllegalStateException
- com/ibm/websphere/ems/InputPort
- com/ibm/websphere/ems/OutputPort
- com/ibm/websphere/ems/transport/jms/JMSRequest
- com/ibm/websphere/ems/TimeoutException
- com/ibm/websphere/ems/TransportException
- com/ibm/ws/spi/ems/CMMFactory
- com/ibm/ws/spi/ems/format/cmm/CMMFormatter
- com/ibm/ws/spi/ems/format/cmm/CMMParser
- com/ibm/ws/spi/ems/format/Formatter
- com/ibm/ws/spi/ems/format/Parser
- com/ibm/ws/spi/ems/transport/CMMReceiver
- com/ibm/ws/spi/ems/transport/CMMReplySender
- com/ibm/ws/spi/ems/transport/CMMSender
- com/ibm/ws/spi/ems/transport/MessageFactory

Recommended migration action:

Instead of using the Extended Messaging Service and its associated tools, you will need to use standard JMS APIs, or equivalent messaging technologies.

Deprecated features in WebSphere Business Integration Server Foundation version 5.1.1

WebSphere Business Integration Server Foundation version 5.1.1 has no deprecated features.

Deprecated features in WebSphere Business Integration Server Foundation version 5.1

Installation and migration tools

Business processes modeled with WebSphere Studio Application Developer Integration Edition version 5.0 or earlier are deprecated.

Recommended migration action:

Use the **Migrate** option provided with WebSphere Studio Application Developer Integration Edition version 5.1 to migrate business process to a BPEL-related process.

Several Business Process Choreographer API interfaces and methods used for business processes created with WebSphere Studio Application Developer Integration Edition version 5.0 or earlier.

Recommended migration action:

See the Javadoc provided with Business Process Choreographer for a detailed list of these API interfaces and methods.

Application programming model and container support features

Business Rule Bean programming interfaces that include the following public classes, methods, and attributes are deprecated:

- Public classes:
 - com.ibm.websphere.brb.RuleImporter
 - com.ibm.websphere.brb.RuleExporter
- Public method:
 - getLocalRuleManager() on class com.ibm.websphere.brb.TriggerPoint
- Protected attribute:
 - ruleMgr on class com.ibm.websphere.brb.TriggerPoint

Recommended migration action:

No action is required.

The com.ibm.websphere.scheduler class programming interface scheduler.Scheduler methods are deprecated:

```
public BeanTaskInfo createBeanTaskInfo();  
public MessageTaskInfo createMessageTaskInfo();
```

Recommended migration action:

Use the following methods:

```
public Object createTaskInfo(Class taskInfoInterface) throws TaskInfoInvalid;  
BeanTaskInfo ti = (BeanTaskInfo) Scheduler.createTaskInfo(BeanTaskInfo.class);
```

The Web Services gateway customization API is deprecated.

Recommended migration action:

No action is required. However, where possible, use the Java API for XML-based Remote Procedure Call (JAX-RPC) handlers rather than Web Services gateway-specific interfaces, such as filters. The Web Services gateway API will be replaced in a future release. For more information, see the topic 'JAX-RPC handlers - An alternative to gateway filters' in the WebSphere Business Integration Server Foundation information center.

Chapter 4. Troubleshooting migration

If you encounter problems during migration, the information described here could help.

Troubleshooting version-to-version migration

Review this page for troubleshooting tips if you encounter problems while migrating from an older version of WebSphere Process Server.

- If you encounter a problem when you are migrating from a previous version of WebSphere Process Server to version 6.1, check your log files and other available information.
 1. Look for diagnostic information in the two files that were generated when you ran the migration job. The files were written to the JESOUT data set during the WROUT and WRERR steps; browse them from SDSF.
 2. Look in the following log files, which are in ASCII format (if you are viewing them on z/OS, you must convert them to EBCDIC first):
 - *migration_backup_directory*/base_backup/WASPreUpgrade.time_stamp.log
 - *migration_backup_directory*/base_backup/WBIPostUpgrade.time_stamp.log
 - *migration_backup_directory*/base_backup/WBIPreUpgrade.time_stamp.log
 - /WebSphere/V6R1/DeploymentManager/profiles/default/logs/WASPreUpgradeSummary.log
 - /WebSphere/V6R1/DeploymentManager/profiles/default/logs/WASPostUpgradeSummary.log
 - /WebSphere/V6R1/DeploymentManager/profiles/default/logs/WASPostUpgrade.time_stamp.log
 - /WebSphere/V6R1/AppServer/profiles/default/logs/WASPreUpgradeSummary.log
 - /WebSphere/V6R1/AppServer/profiles/default/logs/WASPostUpgradeSummary.log
 - /WebSphere/V6R1/AppServer/profiles/default/logs/WASPostUpgrade.time_stamp.log
 3. Look for the following messages in the logs:

MIGR0259I: The migration has successfully completed.

MIGR0271W: The migration completed with warnings.
 4. Open the Log and Trace Analyzer built into the Application Server Toolkit (AST) on the service log of the server that is hosting the resource that you are trying to access, and use it to browse error and warning messages.

See Debugging components in the Application Server Toolkit.
 5. With WebSphere Process Server, run the dumpNameSpace command and pipe, redirect, or "more" the output so that it can be easily viewed.

This command results in a display of all objects in WebSphere Process Server namespace, including the directory path and object name.
 6. If the object a client needs to access does not appear, use the administrative console to verify the following conditions.
 - The server hosting the target resource is started.

- The Web module or Enterprise JavaBean container hosting the target resource is running.
- The JNDI name of the target resource is properly specified.

If none of these steps solves the problem, see Troubleshooting and support for additional troubleshooting resources, including information about how to contact IBM Support.

- During the migration process, problems might occur during the WBIPreUpgrade or WBIPostUpgrade steps.
 - Problems can occur during the WBIPreUpgrade step.
 - A "Not found" or "No such file or directory" message is returned.
This problem can occur if the WBIPreUpgrade script is not in the correct version 6.1 bin directory; for example: /WebSphere/V6R1/DeploymentManager/profiles/default/bin. Verify that the WBIPreUpgrade script is in the correct directory so that the migration job can run the script.
 - The DB2 JDBC driver and DB2 JDBC driver (XA) cannot be found in the drop-down list of supported JDBC providers in the administrative console.
The administrative console no longer displays deprecated JDBC provider names. The new JDBC provider names used in the administrative console are more descriptive and less confusing. The new providers will differ only by name from the deprecated ones.
The deprecated names will continue to exist in the jdbc-resource-provider-templates.xml file for migration reasons (for existing JACL scripts for example); however, you are encouraged to use the new JDBC provider names in your JACL scripts.
 - You receive the following message:
MIGR0108E: The specified WebSphere directory does not contain a WebSphere version that can be upgraded.
This can occur if an incorrect directory was used in the WBIPreUpgrade step of the migration job.
 - Problems can occur during the WBIPostUpgrade step.
 - A "Not found" or "No such file or directory" message is returned.
This problem can occur if the WBIPostUpgrade script is not in the correct version 6.1 bin directory; for example: /WebSphere/V6R1/DeploymentManager/profiles/default/bin. Verify that the WBIPostUpgrade script is in the correct directory so that the migration job can run the script.
 - When you migrate the federated nodes in a cell, you receive the following error messages:
MIGR0304I: The previous WebSphere environment is being restored.
com.ibm.websphere.management.exception.RepositoryException:
com.ibm.websphere.management.exception.ConnectorException: ADMC0009E:
The system failed to make the SOAP RPC call: invoke
MIGR0286E: The migration failed to complete.
A connection timeout occurs when the federated node tries to retrieve configuration updates from the deployment manager during the WBIPostUpgrade migration step for the federated node. Copying the entire configuration might take more than the connection timeout if the configuration that you are migrating to version 6.1 contains any of the following elements:
 - Many small applications
 - A few large applications
 - One very large application

If this occurs, modify the timeout value before running the migration job.

1. Change to the properties directory for the version 6.1 profile to which you are migrating your federated node; for example:
`/WebSphere/V6R1/AppServer/profiles/default/properties`
2. Open the `soap.client.props` file in that directory and find the value for the `com.ibm.SOAP.requestTimeout` property. This is the timeout value in seconds. The default value is 180 seconds.
3. Change the value of `com.ibm.SOAP.requestTimeout` to make it large enough to migrate your configuration. For example, the following entry would give you a timeout value of a half of an hour:
`com.ibm.SOAP.requestTimeout=1800`

Note: Select the smallest timeout value that will meet your needs. Be prepared to wait for at least three times the timeout that you select—once to download files to the backup directory, once to upload the migrated files to the deployment manager, and once to synchronize the deployment manager with the migrated node agent.

4. Go to the following location in the backup directory that was created by the `WBIPreUpgrade` step of the migration job:
`migration_backup_directory/profiles/default/properties`
 5. Open the `soap.client.props` file in that directory and find the value for the `com.ibm.SOAP.requestTimeout` property.
 6. Change the value of `com.ibm.SOAP.requestTimeout` to the same value that you used in the version 6.1 file.
- You receive the "Unable to copy document to temp file" error message. Here is an example:

```
MIGR0304I: The previous WebSphere environment is being restored.  
com.ibm.websphere.management.exception.DocumentIOException: Unable to copy  
document to temp file:  
cells/sunblade1Network/applications/LARGEApp.ear/LARGEApp.ear
```

Your file system might be full. If your file system is full, clear some space and rerun the `WBIPostUpgrade` command.

- You receive the following message:
- ```
MIGR0108E: The specified WebSphere directory does not contain a WebSphere
version that can be upgraded.
```

The following possible reasons for this error exist:

- An incorrect directory might have been used when running the `WBIPreUpgrade` or `WBIPostUpgrade` steps
- The `WBIPreUpgrade` command was not run.

- You receive the following error message:

```
MIGR0253E: The backup directory migration_backup_directory does not exist.
```

The following possible reasons for this error exist:

- An incorrect backup directory might be specified.

For example, the directory might have been a subdirectory of the version 6.0.x tree that was deleted after the `WBIPreUpgrade` command was run and the older version of the product was uninstalled but before the `WBIPostUpgrade` command was run.

1. Determine whether or not the full directory structure specified in the error message exists.
2. If possible, rerun the `WBIPreUpgrade` command, specifying the correct full migration backup directory.

3. If the backup directory does not exist and the older version it came from is gone, rebuild the older version from a backup repository or XML configuration file.
4. Rerun the WBIPreUpgrade command.

- You decide that you need to run WBIPreUpgrade again after you have already run the WBIPostUpgrade command.

During the course of a deployment manager or a managed node migration, WBIPostUpgrade might disable the old environment. If after running WBIPostUpgrade you want to run WBIPreUpgrade again against the old installation, you must run the migrationDisablementReversal.jacl script located in the old *install\_root*/bin directory. After running this JACL script, your version 6.0.x environment will be in a valid state again, allowing you to run WBIPreUpgrade to produce valid results.

For more information on scripting, see Getting started with scripting. Scripting, as described there, is available for WebSphere Process Server.

- A federated migration fails with message MIGR0405E.

The migration that has taken place on your deployment manager as part of your federated migration has failed. For a more detailed reason for why this error has occurred, open the folder *your\_node\_name\_migration\_temp* located on your deployment manager node under the ...DeploymentManagerProfile/temp directory. For example:

```
/websphere61/procserver/profiles/dm_profile/temp/nodeX_migration_temp
```

The logs and everything else involved in the migration for this node on the deployment manager node are located in this folder. This folder will also be required for IBM support related to this scenario.

- WebSphere Process Server version 6.1 applications are lost during migration.

If any of the version 6.1 applications fail to install during a federated migration, they will be lost during the synchronizing of the configurations. The reason that this happens is that one of the final steps of WBIPostUpgrade is to run a syncNode command. This has the result of downloading the configuration on the deployment manager node and overwriting the configuration on the federated node. If the applications fail to install, they will not be in the configuration located on the deployment manager node. To resolve this issue, manually install the applications after migration. If they are standard version 6.1 applications, they will be located in the *install\_root*/installableApps directory.

To manually install an application that was lost during migration, use the wsadmin command to run the *install\_application\_name.jacl* script that the migration tools created in the backup directory.

See the Wsadmin tool.

- WebSphere Process Server version 6.1 applications fail to install.

Manually install the applications using the wsadmin command after WBIPostUpgrade has completed.

To manually install an application that failed to install during migration, use the wsadmin command to run the *install\_application\_name.jacl* script that the migration tools created in the backup directory.

See the Wsadmin tool, or see the WBIPostUpgrade command.

- If you select the option for the migration process to install the enterprise applications that exist in the version 6.0.x configuration into the new version 6.1

configuration, you might encounter some error messages during the application-installation phase of migration.

The applications that exist in the version 6.0.x configuration might have incorrect deployment information—usually, incorrect XML documents that were not validated sufficiently in previous WebSphere Process Server runtimes. The runtime now has an improved application-installation validation process and will fail to install these malformed EAR files. This results in a failure during the application-installation phase of WBIPostUpgrade and produces an "E:" error message. This is considered a "fatal" migration error.

If migration fails in this way during application installation, you can do one of the following:

- Fix the problems in the version 6.0.x applications, and then remigrate.
- Proceed with the migration and ignore these errors.

In this case, the migration process does not install the failing applications but does complete all of the other migration steps.

Later, you can fix the problems in the applications and then manually install them in the new version 6.1 configuration using the administrative console or an install script.

- After migrating to a version 6.1 cell that contains or interoperates with version 6.0.x nodes that are not at WebSphere Process Server version 6.0.1.3 or later, the cluster function might fail.

When starting these version 6.0.x servers, you might see the following problems:

- You might see a first failure data capture (FFDC) log that shows a `ClassNotFoundException` error message. This exception is thrown from the `RuleEtiquette.runRules` method and looks something like the following example:

```
Exception = java.lang.ClassNotFoundException
Source = com.ibm.ws.cluster.selection.SelectionAdvisor.<init>
probeid = 133
Stack Dump = java.lang.ClassNotFoundException: rule.local.server
at java.net.URLClassLoader.findClass(URLClassLoader.java(Compiled Code))
at com.ibm.ws.bootstrap.ExtClassLoader.findClass(ExtClassLoader.java:106)
at java.lang.ClassLoader.loadClass(ClassLoader.java(Compiled Code))
at java.lang.ClassLoader.loadClass(ClassLoader.java(Compiled Code))
at java.lang.Class.forName1(Native Method)
at java.lang.Class.forName(Class.java(Compiled Code))
at com.ibm.ws.cluster.selection.rule.RuleEtiquette.runRules(RuleEtiquette.java:154)
at com.ibm.ws.cluster.selection.SelectionAdvisor.handleNotification(SelectionAdvisor.java:153)
at com.ibm.websphere.cluster.topography.DescriptionFactory$Notifier.run(DescriptionFactory.java:257)
at com.ibm.ws.util.ThreadPool$Worker.run(ThreadPool.java:1462)
```

- You might see a `java.io.IOException` that looks something like the following example:

```
Exception = java.io.IOException
Source = com.ibm.ws.cluster.topography.DescriptionManagerA.update probeid = 362
Stack Dump = java.io.IOException
at com.ibm.ws.cluster.topography.ClusterDescriptionImpl.importFromStream(ClusterDescriptionImpl.java:916)
at com.ibm.ws.cluster.topography.DescriptionManagerA.update(DescriptionManagerA.java:360)
Caused by: java.io.EOFException
at java.io.DataInputStream.readFully(DataInputStream.java(Compiled Code))
at java.io.DataInputStream.readUTF(DataInputStream.java(Compiled Code))
at com.ibm.ws.cluster.topography.KeyRepositoryImpl.importFromStream(KeyRepositoryImpl.java:193)
```

During migration, version 6.1 cluster information is distributed throughout the cell. WebSphere Process Server version 6.0.x nodes that are not at version 6.0.1.3 or later fail to read this information.

To avoid this problem, upgrade all version 6.0.x nodes that will be contained in or interoperating with a version 6.1 cell to version 6.0.1.3 or later before migrating your deployment managers to version 6.1.

- After you migrate a managed node to version 6.1, the application server might not start.

When you try to start the application server, you might see errors similar to those in the following example:

```
[5/11/06 15:41:23:190 CDT] 0000000a SystemErr R
 com.ibm.ws.exception.RuntimeError:
com.ibm.ws.exception.RuntimeError: org.omg.CORBA.INTERNAL:
 CREATE_LISTENER_FAILED_4
vmcid: 0x49421000 minor code: 56 completed: No
[5/11/06 15:41:23:196 CDT] 0000000a SystemErr R at
com.ibm.ws.runtime.WsServerImpl.bootServerContainer(WsServerImpl.java:198)
[5/11/06 15:41:23:196 CDT] 0000000a SystemErr R at
com.ibm.ws.runtime.WsServerImpl.start(WsServerImpl.java:139)
[5/11/06 15:41:23:196 CDT] 0000000a SystemErr R at
com.ibm.ws.runtime.WsServerImpl.main(WsServerImpl.java:460)
[5/11/06 15:41:23:196 CDT] 0000000a SystemErr R at
com.ibm.ws.runtime.WsServer.main(WsServer.java:59)
[5/11/06 15:41:23:196 CDT] 0000000a SystemErr R at
sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
[5/11/06 15:41:23:196 CDT] 0000000a SystemErr R at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:64)
[5/11/06 15:41:23:197 CDT] 0000000a SystemErr R at
sun.reflect.DelegatingMethodAccessorImpl.invoke
(DelegatingMethodAccessorImpl.java:43)
```

Change the port number at which the managed node's server is listening. If the deployment manager is listening at port 9101 for ORB\_LISTENER\_ADDRESS, for example, the server of the managed node should not be listening at port 9101 for its ORB\_LISTENER\_ADDRESS. To resolve the problem in this example, perform the following steps:

1. On the administrative console, click **Application servers** → *server\_name* → **Ports** → ORB\_LISTENER\_ADDRESS.
  2. Change the ORB\_LISTENER\_ADDRESS port number to one that is not used.
- In a network deployment environment, if the error SRVE0026E: [Servlet Error]-[com/ibm/wbi/servers/brules/BusinessRuleManager]: java.lang.NoClassDefFoundError occurs when you access the Business Rules Manager after migrating, you must manually install the Business Rules Manager application on the deployment target before continuing with normal migration of that node. See Migrating Business Rules Manager in a network deployment environment for more information.
  - If synchronization fails when you migrate a managed node to version 6.1, the server might not start.

You might receive messages similar to the following when you migrate a managed node to version 6.1:

```
ADMU0016I: Synchronizing configuration between node and cell.
ADMU0111E: Program exiting with error:
 com.ibm.websphere.management.exception.AdminException: ADMU0005E:
 Error synchronizing repositories
ADMU0211I: Error details may be seen in the file:
 /opt/WebSphere/61AppServer/profiles/AppSrv02/logs/syncNode.log
MIGR0350W: Synchronization with the deployment manager using the SOAP protocol
 failed.
```

MIGR0307I: The restoration of the previous WebSphere Application Server environment is complete.

MIGR0271W: Migration completed successfully, with one or more warnings.

These messages indicate the following:

- Your deployment manager is at a version 6.1 configuration level.
- The managed node that you are trying to migrate is at a version 6.1 configuration level on the deployment manager's repository (including applications).
- The managed node itself is not quite complete given that you did not complete the syncNode operation.

Perform the following actions to resolve this issue:

1. Rerun the syncNode command on the node to synchronize it with the deployment manager.  
See the syncNode command .
  2. Run the GenPluginCfg command.  
See the GenPluginCfg command .
- If you encounter a problem when you are migrating from a previous version of WebSphere Process Server to version 6.1, check your log files and other available information.
    - If the migration job fails before the WBIPostUpgrade step, re-run the migration job.
    - If the migration job fails in the WBIPostUpgrade step, the configuration of the new 6.1 server is partially updated so re-create (or restore from backup) the new 6.1 server before re-running the migration job.
    - Problems occur with a managed (federated) node migration.

A federated node is the most complex node to migrate because it is essentially two migrations rolled into one. A federated node requires a migration of the node configuration information contained in the deployment manager's master repository as well as the configuration information contained in the federated node. Federated node migration requires an active connection to the deployment manager. If you have security enabled, it is essential that you follow the instructions that were generated when you created the migration jobs. The migration job must be submitted with a WebSphere Administrator's user ID that has been properly configured for obtaining secure connections.

version 6.x node agents might display as not synchronized or not available when you change the deployment manager node name in a mixed cell during migration to the version 6.1 deployment manager. version 6.x node agents maintain a link to the version 6.x deployment manager until they are restarted; therefore, they might fail to synchronize with the new deployment manager. The discovery problem, which prevents automatic synchronization, occurs because the node agent is not yet aware of the deployment manager name change that occurred during the migration. If you experience this problem, perform these steps on the node.

      1. Stop the node.
      2. Run the **syncNode** command.
      3. Restart the node.
    - Job fails during the application-installation phase of migration.

If you select the option for the migration process to install the enterprise applications that exist in the version 6.0.x configuration into the new version

6.1 configuration, you might encounter error messages during the application-installation phase of migration.

The applications that exist in the version 6.0.x configuration might have incorrect deployment information—usually, invalid XML documents that were not validated sufficiently in previous WebSphere Process Server runtimes. The runtime now has an improved application-installation validation process and will fail to install these malformed EAR files. This results in a failure during the application-installation phase of WBIPostUpgrade and produces an "E:" error message. This is considered a "fatal" migration error.

If migration fails in this way during application installation, you can do one of the following:

- Fix the problems in the version 6.0.x applications, and then remigrate.
- Proceed with the migration and ignore these errors.
  1. Restart the migration job in the FINISHUP step to allow the remaining migration functions to be performed.

Do this by adding the RESTART=FINISHUP parameter to the job card and resubmitting the job.
  2. Later, fix the problems in the applications and then manually install them in the new version 6.1 configuration using the administrative console or an install script.

- Out of space errors occur.

The migration logs are located in *temporary\_directory\_location/nnnnn*, where *temporary\_directory\_location* is the value that you specified when you created the migration jobs (where the default is */tmp/migrate*) and *nnnnn* is a unique number that was generated during the creation of your migration jobs.

Normally, the space requirements for the migration logs are small. If you enable tracing, however, the log files can be quite large. The best practice is to enable tracing only after problems have been found. If tracing is required, try to only enable tracing related to the step in the process that is being debugged. This will help to reduce the space requirements.

Tracing can be enabled when you create the migration jobs or by changing the variables in the migration JCL from disabled to enabled:

```
TraceState=enabled
profileTrace=disabled
preUpgradeTrace=disabled
postUpgradeTrace=enabled
```

During migration, a backup copy of your version 6.0.x configuration is made. This backup becomes the source of the information being migrated. The default backup location is */tmp/migrate/nnnnn*. This location can be changed when you create the migration jobs. Depending on the size of the node being migrated, this backup can be quite large. If your temporary space is inadequate, then you will need to relocate this backup.

- Batch job time is exceeded.

Each z/OS installation is different with respect to job classes and time limitations. Make sure you have specified appropriate job classes and timeout values on your job card.

- Failures occur during server startup after migration.

Review the instructions that were generated when you created the migration jobs. Verify that the JCL procedures have been copied over correctly to your PROCLIB, the RACF definitions have been created, the version 6.1 libraries have been authorized, and, if required, your STEPLIB statements to the version 6.1 libraries have been specified. Make sure that the daemon process

associated with your cell is at the appropriate level. The daemon process must be at the highest WebSphere Process Server for z/OS version level of all servers that it manages within the cell.

After migrating to a version 6.1 cell that contains or interoperates with version 6.0.x nodes that are not at version 6.0.1.3 or later, the cluster function might fail. When starting these version 6.0.x application servers, you might see the following problems:

- You might see a first failure data capture (FFDC) log that shows a `ClassNotFoundException` error message. This exception is thrown from the `RuleEtiquette.runRules` method and looks something like the following example:

```
Exception = java.lang.ClassNotFoundException
Source = com.ibm.ws.cluster.selection.SelectionAdvisor.<init>
probeid = 133
Stack Dump = java.lang.ClassNotFoundException: rule.local.server
at java.net.URLClassLoader.findClass(URLClassLoader.java(Compiled Code))
at com.ibm.ws.bootstrap.ExtClassLoader.findClass(ExtClassLoader.java:106)
at java.lang.ClassLoader.loadClass(ClassLoader.java(Compiled Code))
at java.lang.ClassLoader.loadClass(ClassLoader.java(Compiled Code))
at java.lang.Class.forName1(Native Method)
at java.lang.Class.forName(Class.java(Compiled Code))
at com.ibm.ws.cluster.selection.rule.RuleEtiquette.runRules
(RuleEtiquette.java:154)
at com.ibm.ws.cluster.selection.SelectionAdvisor.handleNotification
(SelectionAdvisor.java:153)
at com.ibm.websphere.cluster.topography.DescriptionFactory$Notifier.run
(DescriptionFactory.java:257)
at com.ibm.ws.util.ThreadPool$Worker.run(ThreadPool.java:1462)
```

- You might see a `java.io.IOException` that looks something like the following example:

```
Exception = java.io.IOException
Source = com.ibm.ws.cluster.topography.DescriptionManagerA.update probeid
= 362
Stack Dump = java.io.IOException
at com.ibm.ws.cluster.topography.ClusterDescriptionImpl.importFromStream
(ClusterDescriptionImpl.java:916)
at com.ibm.ws.cluster.topography.DescriptionManagerA.update
(DescriptionManagerA.java:360)
Caused by: java.io.EOFException
at java.io.DataInputStream.readFully(DataInputStream.java(Compiled Code))
at java.io.DataInputStream.readUTF(DataInputStream.java(Compiled Code))
at com.ibm.ws.cluster.topography.KeyRepositoryImpl.importFromStream
(KeyRepositoryImpl.java:193)
```

During migration, version 6.1 cluster information is distributed throughout the cell. Version 6.0.x nodes that are not at version 6.0.1.3 or later fail to read this information. To avoid this problem, upgrade all version 6.0.x nodes that will be contained in or interoperating with a version 6.1 cell to version 6.0.1.3 or later before migrating your deployment managers to version 6.1.

After migration, carefully review the job output and log files for errors.

**Note:** WebSphere Process Server provides an interactive problem control system (IPCS) verb `exit` to help you to format information from dumps of WebSphere Process Server processes. This verb `exit` was named `CBDATA`, which was an alias of the real module name, in version 6.0.x and earlier. In version 6.1, that alias was removed. In version 6.1 and later, therefore, you must use the real name of this verb `exit`, `BBORDATA`, instead of the alias.

If you migrate a node to version 6.1 then discover that you need to revert back to version 6.0.x, see [Rolling back your environment](#).

If none of these steps solves the problem, see [Troubleshooting and support](#) for additional troubleshooting resources, including information about how to contact IBM Support.

- In a network deployment environment, if the error SRVE0026E: [Servlet Error]-[com/ibm/wbi/servers/brules/BusinessRuleManager]: java.lang.NoClassDefFoundError occurs when you access the Business Rules Manager after migrating, you must manually install the Business Rules Manager application on the deployment target before continuing with normal migration of that node. See [Migrating Business Rules Manager in a network deployment environment](#) for more information.

## What to do next

If you did not find your problem listed, contact IBM support.

### Related concepts

“Migration considerations for Business Process Choreographer” on page 30  
If your servers run Business Process Choreographer, you should be aware of some restrictions and additional tasks you might need to perform.

### Related tasks

“Verifying migration” on page 23

Verify that your migration was successful by checking the log files and checking operation with the administrative console.

### Related reference

 [WBIPreUpgrade command](#)

Use the WBIPreUpgrade command for WebSphere Process Server to save the configuration of a previously installed version of WebSphere Process Server into a migration-specific backup directory.

 [WBIPostUpgrade command](#)

Use the WBIPostUpgrade command for WebSphere Process Server to retrieve the saved configuration that was created by the WBIPreUpgrade command from the *backupDirectory* that you specified. The WBIPostUpgrade command for WebSphere Process Server reads the configuration from this directory to migrate to the more recent version of WebSphere Process Server and adds all migrated applications into the *profile\_root/installedApps* directory for the new installation.

### Related information

[Debugging components in the Application Server Toolkit](#)

[Wsadmin tool](#)

[syncNode command](#)

[GenPluginCfg command](#)

 [Troubleshooting and support](#)

To help you understand, isolate, and resolve problems with your IBM software, the troubleshooting and support information contains instructions for using the problem-determination resources that are provided with your IBM products.

[Getting started with scripting](#)



---

## Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

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

*IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.*

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*IBM World Trade Asia Corporation Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106-0032, Japan*

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
577 Airport Blvd., Suite 800  
Burlingame, CA 94010  
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows: (c) (your company name) (year). Portions of

this code are derived from IBM Corp. Sample Programs. (c) Copyright IBM Corp. \_enter the year or years\_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

## **Programming interface information**

Programming interface information, if provided, is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

**Warning:** Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

## **Trademarks and service marks**

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (<sup>®</sup> or <sup>™</sup>), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

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

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are registered 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, or service names may be trademarks or service marks of others.

This product includes software developed by the Eclipse Project (<http://www.eclipse.org>).



IBM WebSphere Process Server for z/OS, Version 6.1.2





Printed in USA