WebSphere® Process Server for Multiplatforms

Version 6.2.0

Migrating WebSphere Process Server

WebSphere® Process Server for Multiplatforms

Version 6.2.0

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.

24 April 2009

This edition applies to version 6, release 2, modification 0 of WebSphere Process Server for Multiplatforms (product number 5724-L01) 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.

PDF books and the information center

PDF books are provided as a convenience for printing and offline reading. For the latest information, see the online information center.

As a set, the PDF books contain the same content as the information center.

The PDF documentation is available within a quarter after a major release of the information center, such as Version 6.0 or Version 6.1.

The PDF documentation is updated less frequently than the information center, but more frequently than the Redbooks®. In general, PDF books are updated when enough changes are accumulated for the book.

Links to topics outside a PDF book go to the information center on the Web. Links to targets outside a PDF book are marked by icons that indicate whether the target is a PDF book or a Web page.

Table 1. Icons that prefix links to topics outside this book

Icon	Description

➡	A link to a Web page, including a page in the information center.
	Links to the information center go through an indirection routing service, so that they continue to work even if target topic is moved to a new location.
	If you want to find a linked page in a local information center, you can search for the link title. Alternatively, you can search for the topic id. If the search results in several topics for different product variants, you can use the search result **Group by** controls to identify the topic instance that you want to view. For example:
	1. Copy the link URL; for example, right-click the link then select Copy link location. For example: `http://www14.software.ibm.com/webapp/wsbroker/redirect?version=wbpm620&product=wesb-dist&topic=tins_apply_service`
	2. Copy the topic id after `&topic=`. For example: `tins_apply_service`
	3. In the search field of your local information center, paste the topic id. If you have the documentation feature installed locally, the search result will list the topic. For example:
	`1 result(s) found for`
	`Group by: None | Platform | Version | Product`
	`Show Summary`
	`Installing fix packs and refresh packs with the Update Installer`
	4. Click the link in the search result to display the topic.

📄	A link to a PDF book.

Contents

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 6.2.

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. Refresh packs, interim fixes, and fix packs are examples of updates. For more information on updating, see Installing fix packs and refresh packs with the Update Installer.

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. For example, if your WebSphere Process Server version 6.1.x is running on Microsoft® Windows® XP, you can migrate the data from that instance of WebSphere Process Server to WebSphere Process Server version 6.2 running on Windows XP. However, you could not migrate data from WebSphere Process Server version 6.1.x running on Windows XP to WebSphere Process Server version 6.2 running on AIX®. In the case of a standalone server only, you can migrate from a prior release of a given operating system to a new supported release of the same operating system. (Refer to "Migrating from an operating system that is no longer supported" on page 64 for instructions on such a migration.)

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 6.2.

Table 2. Supported version-to-version migration scenarios

Currently installed version	New version
WebSphere Process Server version 6.0.2.x	WebSphere Process Server version 6.2
WebSphere Process Server version 6.1.x	WebSphere Process Server version 6.2
WebSphere Enterprise Service Bus version 6.0.2.x	WebSphere Process Server version 6.2
WebSphere Enterprise Service Bus version 6.1.x	WebSphere Process Server version 6.2

Note: If you are migrating from a version of WebSphere Process Server that is earlier than version 6.0.2.x, you must first migrate to version 6.0.2.x, then migrate to version 6.2. For information about migrating to WebSphere Process Server version 6.0.2.x, refer to the WebSphere Process Server version 6.0.2.x information center.

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 WebSphere Process Server or WebSphere Application Server profiles, 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.

For some releases of WebSphere Process Server, an ″in-place update″ or maintenance package is available. Such an update will also preserve configuration data. In cases for which no maintenance package is available, such as when moving from WebSphere Process Server version 6.1.x or version 6.0.2.x to version 6.2, a version-to-version migration is required to preserve your configuration data.

Profile types: WebSphere Process Server, WebSphere Enterprise Service Bus, and WebSphere Application Server profiles

You can migrate three types of profiles to WebSphere Process Server version 6.2: WebSphere Process Server, WebSphere Enterprise Service Bus, and WebSphere Application Server profiles.

Important: Even though the following definitions refer to the Profile Management Tool as a tool that may have been used to create the profiles being migrated, you cannot use the Profile Management Tool to create new target profiles during migration. During migration, you use only the migration wizard or the command-line tools to create new target profiles.

WebSphere Process Server profile

A WebSphere Process Server profile is one that was created using one of the following profile templates: ″default.wbiserver,″ ″dmgr.wbiserver,″ or ″managed.wbiserver.″ When using the Profile Management Tool, this means selecting **WebSphere Process Server** on the Environment selection page.

WebSphere Enterprise Service Bus

A WebSphere Enterprise Service Bus profile is one that was created using one of the following profile templates: ″default.esbserver,″ ″dmgr.esbserver,″ or ″managed.esbserver.″ When using the Profile Management Tool, this means selecting **WebSphere Enterprise Service Bus** on the Environment selection page.

WebSphere Application Server profile

A WebSphere Application Server profile is one that was created using one of the following profile templates: ″default″, ″dmgr,″ or ″managed.″ When using the Profile Management Tool, this means selecting one of the following choices on the Environment selection page: **Application Server**, **Deployment Manager**, or **Custom profile**.

Note: In a WebSphere Process Server network deployment environment, the deployment manager must always be created in a WebSphere Process Server profile.

Note: In a WebSphere Process Server client installation, only WebSphere Application Server profiles can be created.

Table 3. Profiles supported during migration

Profile type	Environment in which profile was created	Can profile be migrated using the migration wizard?	Can profile be migrated using command-line tools?
WebSphere Process Server	WebSphere Process Server - server installation **Note:** This includes the WebSphere Process Server client installation option	Yes	Yes
WebSphere Enterprise Service Bus	WebSphere Process Server - server installation **Note:** This includes the WebSphere Process Server client installation option	Yes	Yes
WebSphere Application Server	WebSphere Process Server - client installation	No	Yes
WebSphere Application Server	WebSphere Process Server - server installation	No	Yes

Related concepts

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

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.

Related tasks

Installing fix packs and refresh packs with the Update Installer
You can use the IBM® Update Installer for WebSphere Software to install interim fixes, fix packs, and refresh packs collectively known as maintenance packages. The Update Installer for WebSphere Software is also known as the update installer program, the UpdateInstaller program, and the Update Installation Wizard.

Premigration considerations for WebSphere Process Server

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, restrictions, and considerations apply to migration and coexistence if you have WebSphere Process Server version 6.2 installed.

- “WebSphere Process Server installation requirements”
- “Augmentation”
- “Backup directory” on page 5
- “Cell configuration” on page 5
- “Clusters” on page 5
- “Databases” on page 5
- “HTTP transport” on page 5
- “Java/JDK (Java Development Kit)” on page 6
- “JNI (Java Native Interface)” on page 6
- “Migration tools” on page 6
- “Profiles” on page 7
- “Rolling back environments” on page 7
- “Storage” on page 7
- “Ulimit setting” on page 8

WebSphere Process Server installation requirements

- WebSphere Process Server version 6.2 can be installed in an environment where it coexists with prior levels of WebSphere Process Server. However, some restrictions exist.

  For information about coexistence, including restrictions, see Coexisting with other WebSphere product installations.

Augmentation

- You can migrate a version 6.0.2.x or version 6.1.x profile to a version 6.2 profile only if they are at the same augmentation level.

- You can have a mixed cell containing both augmented and unaugmented managed nodes as long as the deployment manager for the cell 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 been augmented only for WebSphere Application Server can manage only WebSphere Application Server nodes.

Backup directory

- 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 profile's configuration directory and applications. The previous profile can be either a WebSphere Process Server or WebSphere Application Server profile.

  Note: When you migrate from version 6.0.2.x, all of the existing profiles under the previous installation of WebSphere Process Server are backed up. However, when you migrate from version 6.1.x, only one profile at a time is backed up.

Cell configuration

- When migrating a deployment manager or a federated node, the WebSphere Process Server version 6.2 cell name must match the version 6.1.x or 6.0.2.x cell name.

  If you create a WebSphere Process Server or WebSphere Application Server profile with a new cell name and use this profile as a migration target, the migration will fail.

Clusters

- Members of a cluster cannot be running different versions (6.0.2.x, 6.1.x, 6.2) of WebSphere Process Server. If you have configured a cluster containing servers running different versions, all the members running earlier versions of WebSphere Process Server must be stopped before you start the first version 6.2 cluster member. Also, once you start a version 6.2 cluster member, do not start any version 6.1.x or 6.0.2.x cluster members in that cluster.

Databases

- Before you migrate a Cloudscape® or Derby database, ensure that any servers hosting applications that are using the Cloudscape database are shut down. Otherwise, the Cloudscape migration will fail.

HTTP transport

WebSphere Process Server version 6.0.2.x migration converts HTTP transports to channel-framework Web container transport chains.

Note: This applies only to migrations from version 6.0.2.x.

For more information on version 6.2 transport support, see the following topics:

- Configuring transport chains
- HTTP transport channel settings
- Transport chains

Java/JDK (Java™ Development Kit)

- When migrating from version 6.0.2.x, and before you migrate to JDK 5 (introduced in WebSphere Application Server version 6.1 and therefore WebSphere Process Server version 6.2) from JDK 1.4, review your applications for necessary changes based on the Sun Microsystems Java specification.

  Note: This task is not necessary if you are migrating from version 6.1.x.

  See API and specification migration.
- When migrating a cell with multiple nodes, the applications must remain at the lowest JDK level until all nodes are migrated.

Important: Ensure that all generic JVM (Java Virtual Machine) parameters specified for any server are compatible with the new Java version. Remove them if they aren't. Check other JVM related applications such as Wily Agents, because you may have to adopt them to the new Java version. Be sure to disable them prior to the migration and then enable them after the migration.

JNI (Java Native Interface)

- Solaris Java Native Interface (JNI) applications that work with WebSphere Process Server version 6.0.2 on Solaris x64 must be recompiled in a 64-bit environment in order for them to work with WebSphere Process Server version 6.2. This includes all JNI applications that run in a WebSphere Process Server process—code called from an Enterprise JavaBean (EJB) for example.

  On Solaris x64, WebSphere Process Server version 6.0.2 runs as a 32-bit application even though the underlying platform is 64-bit. This is because the underlying Java virtual machine is 32-bit. WebSphere Process Server version 6.2 runs as a 64-bit application because the underlying Java virtual machine is 64-bit. JNI applications compiled in a 32-bit environment for version 6.0.2 cannot run in the 64-bit environment of version 6.2.

Migration tools

Note: This applies only to migrations from version 6.0.2.x.

- After you use the migration tools to migrate to WebSphere Process Server version 6.2, 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 WebSphere Application Server version 6.0.2.x, and make sure that WebSphere Process Server version 6.2 security is set appropriately.

    See Lightweight Third Party Authentication.
  - Check the WBIPostUpgrade.log file in the logs directory for details about any JSP objects that the migration tools did not migrate.

    If WebSphere Application Server does not support a level for which JSP objects are configured, the migration tools recognize the objects in the output and logs them.
  - Review your Java virtual machine settings to verify that you are using the recommended default values as described in

    Java virtual machine settings.
  - Verify the results of the automatic Cloudscape database migration, and manually migrate any Cloudscape databases that are not automatically migrated by the tools.

See “Migrating Cloudscape databases” on page 128.

Profiles

- For information about creating target profiles, refer to “Target profile considerations” on page 9.
- For information about profile types, refer to “Overview of migrating” on page 1.

Rolling back environments

- If you migrate a node to WebSphere Process Server version 6.2, then discover that you need to revert back to version 6.1.x or 6.0.2.x, see “Rolling back your environment” on page 117.

Storage

- The amount of storage that your system requires during migration to version 6.2 depends on your environment as well as on which migration tool you are using.
  - **WBIPreUpgrade storage requirements**
    - **Location:** Backup directory specified as a parameter of the WBIPreUpgrade command
    - **Amount:** For a rough estimate of your storage requirements when using this command, add the following amounts.
      - Size of the following items for all of the WebSphere Process Server or WebSphere Application Server profiles in your old configuration:
        - *profile_root*/installableApps directory
        - *profile_root*/installedApps directory
        - *profile_root*/config directory
        - *profile_root*/properties directory
        - Shared libraries referenced in the libraries.xml configuration files
        - Resource Adapter Archive (RAR) files referenced in the resources.xml configuration files
      - If trace is enabled, which is the default, up to 200 MB (depending on the size and complexity of your configuration)

    For more information about this command, see the WBIPreUpgrade command-line utility.
  - **WBIPostUpgrade storage requirements**
    - **Location:** New configuration relative to the new *profile_root* directory
    - **Amount:** For a rough estimate of your storage requirements when using this command, add the following amounts.
      - Size of the following items for the old WebSphere Process Server or WebSphere Application Server profile that you are migrating:
        - *profile_root*/installableApps directory
        - *profile_root*/installedApps directory
        - *profile_root*/config directory
        - *profile_root*/properties directory
        - Shared libraries referenced in the libraries.xml configuration files
        - Resource Adapter Archive (RAR) files referenced in the resources.xml configuration files
      - If trace is enabled, which is the default, up to 200 MB (depending on the size and complexity of your configuration)

For more information about this command, see the WBIPostUpgrade command-line utility.

Linux UNIX

Ulimit setting

- To avoid an error during post migration where there are too many open files, make sure you increase the ulimit setting. For instructions on increasing the ulimit setting, see Preparing Linux systems.

Related concepts

Premigration considerations for Business Process Choreographer
If your servers run Business Process Choreographer, you should be aware of certain things that you need to plan and take into consideration before you migrate Business Process Choreographer.

Related tasks

Rolling back your environment
After migrating to a WebSphere Process Server version 6.2 environment, you can roll back to a version 6.1.x or 6.0.2.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.

Migrating Cloudscape databases
After you use the migration tools to migrate to WebSphere Process Server version 6.2, you should verify the results of the automatic Cloudscape database migration and manually migrate any Cloudscape database instances that are not automatically migrated by the tools.

Related reference

WBIPreUpgrade command-line utility
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-line utility
Use the WBIPostUpgrade command for WebSphere Process Server to retrieve the profile configuration that was saved by the WBIPreUpgrade command at the *backupDirectory* that you had specified.

Related information

Coexisting with other WebSphere product installations
An installation of WebSphere Process Server, version 6.2 can coexist on the same system with installations of any version of WebSphere Enterprise Service Bus, WebSphere Process Server, 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

Target profile considerations

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

Creating the correct target profile for migration is an important part the migration process. It is strongly recommend that you created new target profiles when migrating your profiles. When you create a migration target profile, you must create it with the same node name, cell name, and augmentation levels as the WebSphere Process Server version 6.1.x or 6.0.2.x profile or WebSphere Application Server profile, and if possible, the same host name.

Tip: If you have created your target profile beforehand, do not make any cell-level changes to the new version 6.2 node before migration, such as changes to virtual-host information, because these changes will be lost during migration. Instead, wait until after the node has been migrated before making any such changes. Otherwise, you will have to use the administrative console running on the deployment manager to manually remake all of the changes to the new cell after migration, such as any changes to the virtual-host and host-alias information.

The migration wizard and the command-line tools both contain built-in options to facilitate the creation of new target profiles.

Migration wizard

If you are using the migration wizard, regardless of what type of profiles you are migrating from, always choose the **Create new profile** option on the migration wizard panel to create a new profile in WebSphere Process Server version 6.2 for migration. This option creates a new profile with the correct configuration settings for migration purposes.

Note: The migration wizard does not support migrating WebSphere Application Server profiles. You must use the command-line tools when migrating WebSphere Application Server profiles in full WebSphere Process Server installations or in WebSphere Process Server client installations.

Note: The migration wizard cannot run in a non-graphical environment. Examples of non-graphical environments include the i5/OS platform or telnet sessions. If you want to run migration in a non-graphical environment, use the command-line tools.

Command-line tools

If you are using the migration command-line tools, regardless of what type of profiles you are migrating from, use **-createTargetProfile** when invoking WBIPostUpgrade to create a new profile in the WebSphere Process Server version 6.2 installation for migration purposes. For a full description of the migration command-line tools, refer to WBIPreUpgrade command-line utility and WBIPostUpgrade command-line utility.

Note: You can optionally use the manageprofiles command if you want to create target profiles for migration outside of the migration framework. An example is if you are preparing target profiles when a migration task is running.

Note: i5/OS If you are migrating on an i5/OS platform, you should be aware of the following considerations:

- The target profile name must match the profile name of the source profile being migrated.
- If you specify `-createTargetProfile` on the WBIPostUpgrade script, you may invoke the *install_root*/bin/enableJVM script to set the target profile JVM type to

the desired value after the target profile has been migrated. The WBIPostUpgrade script ensures that the target profile is generated with the same JVM type as the source profile for the migration process.

- If you manually generate the target profile using `manageprofiles`, you must ensure that the target profile JVM type matches the source profile JVM type. This can be done by invoking the *install_root*/bin/enableJVM script to set the target profile JVM type before invoking the WBIPostUpgrade script.
- The [-profile *target_profile_name*] option of the enableJVM command must be used to prevent switching all profiles in the product install to the specified type.

Examples of profiles created using command-line tools

Note: In the following examples, use one of the following parameters for `-templatePath`, depending on the type of profile you are creating:

- **WebSphere Process Server stand-alone server**: `<WAS_HOME>/profileTemplates/default.wbiserver`
- **WebSphere Process Server deployment manager**: `<WAS_HOME>/profileTemplates/dmgr.wbiserver`
- **WebSphere Process Server custom nodes**: `<WAS_HOME>/profileTemplates/managed.wbiserver`
- **WebSphere Application Server stand-alone server**: `<WAS_HOME>/profileTemplates/default`
- **WebSphere Application Server custom nodes**: `<WAS_HOME>/profileTemplates/managed`

Example: Creating WebSphere Process Server profiles

- **Stand-alone server and deployment manager**: `manageprofiles -create -profileName <profname> -templatePath <example: <WAS-HOME>/profileTemplates/(default)(dmgr).wbiserver> -profilePath <example: /QIBM/UserData/<WAS_HOME>/profiles/<profname>> -winserviceCheck false -defaultPorts -createDefaultProfileForMigration true -dbDelayConfig true -omitAction defaultAppDeployAndConfig`
- **Custom nodes**: `manageprofiles -create -profileName <profname> -templatePath <example: <WAS-HOME>/profileTemplates/managed.wbiserver> -nodeName <nodename> -cellName <cellname> -hostName <hostname> -defaultPorts -createDefaultProfileForMigration true -dbDelayConfig true`

Example: Creating WebSphere Application Server profiles

- **Stand-alone server and deployment manager**: `manageprofiles -create -profileName <profname> -templatePath <WAS_HOME>/profileTemplates/default -profilePath <WAS_HOME>/profiles/<profname> -cellName <cellname> -winserviceCheck false -defaultPorts -hostName <hostname> -omitAction defaultAppDeployAndConfig -nodeName <nodename> -enableAdminSecurity false`
- **Custom nodes**: `manageprofiles -create -profileName <profname> -templatePath <WAS_HOME>/profileTemplates/managed -profilePath <WAS_HOME>/profiles/<profname> -cellname <cellname> -hostName <hostname> -nodeName <nodename> -federateLater true`

Related concepts

Overview of migrating

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

Premigration considerations for Business Process Choreographer

If your servers run Business Process Choreographer, you should be aware of certain things that you need to plan and take into consideration before you migrate Business Process Choreographer.

Database: Backup

Before you upgrade the Business Process Choreographer database, you must make a complete backup of your database, because if the data migration fails you might have to restore your database from the backup.

Database: Upgrade and migration scripts

To upgrade the Business Process Choreographer database, you must run two or three scripts in the following sequence:

1. For some database types and versions, you must run a script that upgrades the table spaces.
2. For all databases, you must run a script to upgrade the schema.
3. For all databases, you must run a script to migrate the runtime data to the new schema. For DB2® UDB and DB2 for z/OS®, the script also migrates data to the new table spaces.

For a production system, you must upgrade to the new database schema manually. The database server must be running, but the WebSphere Process Server servers where Business Process Choreographer is configured must not be started until after the data migration is complete.

Database: Authorization

Because each of these scripts requires different database permissions, check whether you will be able to run all scripts using a single user ID, or whether your database administrator might have to run any of them.

To run the upgradeTablespaces SQL script for DB2 for Linux®, UNIX®, and Windows, you require the following permissions:

CREATE BUFFERPOOL and CREATE TABLESPACE .

To run the upgradeTablespaces SQL script for DB2 for z/OS, you require the following permissions:

CREATE TABLESPACE

To run the upgradeSchema SQL script, you require the following permissions:

For all database types, you must be able to perform CREATE TABLE, ALTER TABLE, DROP INDEX, CREATE INDEX, CREATE VIEW, and DROP VIEW .

To run the migrateDB.py script, you require the following permissions:

- For all database types, you must be able to perform SELECT, INSERT, UPDATE, CREATE VIEW, and DROP VIEW.

- If you are using DB2® Universal Database™ for i5/OS®, make sure that you are using a user profile with *ALLOBJ and *SECADM special authorities.
- If you are using DB2 for Linux, UNIX, Windows, or z/OS, your user ID must also have the following rights for the table space migration: CREATE TABLE, RENAME TABLE, CREATE INDEX, DROP INDEX, CREATE VIEW, and DROP VIEW.

If you configured materialized views, you also require DROP TABLE and CREATE TABLE.

Data migration: Time requirements and tuning options

Depending on the quantity of data and the power of your database server, the data migration step (excluding the time required to backup the database and upgrade the database schema) can take several hours. For the latest information about performing the data migration, including tuning options that can reduce the time required, refer to Technote 21327385.

Data migration: Storage requirements

The data migration merges data from old tables into new tables, and temporarily requires extra storage. To be on the safe side, make sure that your database has enough free storage space for a second copy of all the data. After you have verified that the servers can be started, an optional step describes which tables you can delete to free up the storage.

Data migration: Materialized views

If you were using a custom table definition file for named materialized views, they will be dropped by the data migration script. WebSphere Process Server will only be able to re-create the named materialized views if the **customTableDefinition** points to a custom table definition XML file that it can access. To verify that WebSphere Process Server will be able to re-create your named materialized views perform the following:

1. Make sure that your WebSphere Process Server is up and running.
2. In the administrative console, click either **Servers** → **Application servers** → ***serverName*** or **Clusters** → ***clusterName***, then under **Business Integration** expand **Business Process Choreographer**, then click **Business Flow Manager** → **Custom Properties**.
3. In the list of custom properties for the business container, search for an entry named **customTableDefinition** . This entry specifies the file system location of the custom table definition file, for example, *path*/customData.xml.
4. Verify that the XML file exists:
   - In a stand-alone environment, on the server node
   - In a cluster environment, on each node that hosts a cluster member

   Note: If the file system location of the XML file contains a WebSphere variable, for example, *${WAS_INSTALL_ROOT}*, the value of this variable can change during migration. You might need to copy your XML file to the new location before you start your migrated servers or clusters.
5. Make sure that the XML file can be accessed by WebSphere Process Server.

During migration: Mixed cell restrictions

If, during the course of migrating WebSphere Process Server from version 6.1.x to version 6.2, your cell is running nodes concurrently at the new and premigration levels, be aware of the following:

- When a deployment manager has been migrated to the latest version, you cannot perform any of the following actions on nodes in the cell that are still at premigration level:
  - Configure Business Process Choreographer.
  - Install, update, or uninstall any applications that contain business processes, human tasks, or both.
- When you have a cluster at the latest version that has Business Process Choreographer configured on it, you must not create any new cluster members on any nodes that are still at the premigration level.

After migration: Possible side-effect on results returned by the Query API

After both work item tables are merged, the WORK_ITEM_T table contains new entries. All the new entries also have a unique work item identifier (WIID). Therefore it is possible that some queries against the Query API will return a different result. For example, a count of the number of distinct WIIDs in the WORK_ITEM view will probably return a higher number. However, the total number of entries in the WORK_ITEM view is not affected.

Related concepts

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

Related tasks

"Upgrading the Business Process Choreographer database manually" on page 38
After migrating a server or cluster that has Business Process Choreographer configured, you must manually upgrade the schema for the Business Process Choreographer database and perform a data migration before you start the server or any cluster member.

"Upgrading the Business Process Choreographer database schema" on page 39
After migrating or upgrading a server or cluster that has Business Process Choreographer configured, the schema for the associated Business Process Choreographer database must be upgraded.

"Migrating the Business Process Choreographer runtime data" on page 42
After migrating or upgrading a server or cluster that has Business Process Choreographer configured, you must perform a data migration before you start the server or any cluster member.

"Postmigration tasks for Business Process Choreographer" on page 123
If your servers or clusters run Business Process Choreographer, you must perform some additional tasks before you start your servers or clusters.

Related reference

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

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.

Premigration considerations for Business Space powered by WebSphere

Before you begin the process of migrating WebSphere Process Server from version 6.1.2 to version 6.2, you should be aware of these Business Space considerations.

The following rules, restrictions, and considerations apply to Business Space if you are migrating WebSphere Process Server from version 6.1.2 to version 6.2 .

"Mixed nodes"

"Target profile configuration" on page 16

"Migrating multiple products configured with Business Space" on page 16

Mixed nodes

A mixed node migration with one or more clusters configured with Business Space is supported only if the clusters configured with Business Space are migrated last. A mixed node migration is one during which the newer version deployment manager can manage both older and newer versions of WebSphere Process Server. For example, after migration, a version 6.2 deployment manager can manage both version 6.1.2 and version 6.2 nodes. The managed nodes of the previous deployment manager are now running as version 6.1.2 managed nodes in the version 6.2 deployment manager.

The first version of Business Space is version 6.1.2. If you choose to gradually migrate products shipping Business Space from version 6.1.2 to version 6.2, you must migrate the product applications and binaries before migrating Business Space. This order is significant because widget endpoint code provided by the products will work with Business Space version 6.1.2, but Business Space version 6.2 will not work with widget endpoint code shipped with version 6.1.2 of the products. In order to use a mixture of migrated products, other products must be migrated before Business Space.

Target profile configuration

If you are migrating from WebSphere Process Server version 6.1.2 and have Business Space configured, create a target profile without configuring it for Business Space The reason to skip configuring the target profile with Business Space is to avoid creating a new Business Space database. You update the Business Space version 6.1.2 database tables so they can be used by Business Space version 6.2 during steps described in ″Postmigration tasks for Business Space powered by WebSphere.″

To create a target profile without configuring it for Business Space, you can either select **Create new profile** in the migration wizard or use the following parameter when invoking the `WBIPostUpgrade` command: `-createTargetProfile`.

Migrating multiple products configured with Business Space

If you are migrating other products in addition to WebSphere Process Server, such as WebSphere Business Monitor or WebSphere Business Modeler Publishing Server, additional steps may be necessary. For more information, see the migrating documentation for those products.

Target profile considerations for Business Space

If you are using Business Space, you must configure WebSphere Process Server profiles to work with Business Space after migration. If you are running multiple instances of Business Space, you must configure profiles for each instance of Business Space that you want to work with WebSphere Process Server.

The actions needed for adding Business Space differ, depending on the environment you migrated from. This environment is listed as the source profile in the table that follows. The first release of WebSphere Process Server to have Business Space is version 6.1.2.

The following table details the actions required for configuring WebSphere Process Server profiles to work with Business Space after a migration. The actions you are required to take depend on the version of WebSphere Process Server you are migrating from and whether the source profile has Business Space configured.

Table 4. Actions required for configuring profiles for Business Space

Source profile	Action required for configuring new target profiles for Business Space
• Version 6.0.2.x (Business Space does not exist in this version) • Version 6.1.0.x (Business Space does not exist in this version) • Version 6.1.2.x (without Business Space configured)	1. During migration, create the target profile so it is not configured for Business Space. For more information about target profiles, refer to “Target profile considerations” on page 9. 2. Following migration, use the WebSphere Application Server administrative console to configure the migrated profile for Business Space. You do this to add the Business Space applications and create the Business Space version 6.2 database applications and create the Business Space version 6.2 database. For more information, refer to ″Configuring Business Space using the administrative console.″ 3. Manually configure endpoints if needed.
Version 6.1.2.x (with Business Space configured)	1. During migration, create the target profile so it is not configured for Business Space. For more information about target profiles, refer to “Target profile considerations” on page 9. 2. Following migration, perform the postmigration tasks to upgrade the Business Space version 6.1.2 database for use by Business Space version 6.2, and configure endpoints. Information on configuring endpoints can be found at ″Enabling Business Space widget endpoints on the administrative console″ and ″Enabling Business Space widgets manually for remote endpoints.″

Related tasks

Configuring Business Space using the Profile Management Tool

Configuring Business Space using the administrative console

Tools for version-to-version migration

You can perform migration from earlier versions of WebSphere Process Server or WebSphere Enterprise Service Bus by using a migration ″wizard″ (graphical user interface) or a series of commands or scripts.

Migration wizard

The migration wizard steps you through the migration process. You will be asked to fill in certain fields and make some choices provided by the wizard or use the defaults. You can invoke the migration wizard from the WebSphere Process Server First Steps panel. You can also invoke it directly from *install_root*\bin\wbi_migration.bat (on Windows systems) or *install_root*/bin/wbi_migration.sh (on UNIX-based systems).

Note: If you are using an i5/OS operating system, you cannot use the migration wizard. You must use the migration commands.

Note: The migration wizard supports only WebSphere Process Server profiles. If you have WebSphere Application Server profiles, you must use the migration commands.

Migration commands

You can use a series of scripts invoked from the command line to migrate from an earlier version of WebSphere Process Server if you choose not to use the migration wizard. The following commands must be run in the correct sequence:

1. WBIPreUpgrade - This command, which you run first, saves the existing WebSphere Process Server configuration and applications into a migration-specific backup directory.
2. WBIPostUpgrade - This command, which you run second, processes the contents of the migration-specific backup directory that was created with the WBIPreUpgrade command and imports it into the new WebSphere Process Server environment.
3. WBIProfileUpgrade.ant - This script needs to be run manually after running the WBIPreUpgrade and WBIPostUpgrade commands only if you are migrating a cell containing clusters. You might also need to run it in special cases in which migration did not occur successfully. The script updates enterprise applications and configuration settings in a profile. The WBIPostUpgrade command invokes this script and for non-clustered scenarios does not have to be manually re-run a second time.

Database upgrade scripts

Normally, if updates are required to any databases supporting WebSphere Process Server components, the update is completed automatically when the migrated server process is started. However, some circumstances require that the databases be upgraded manually.

You must update the databases manually using scripts provided with WebSphere Process Server in the following circumstances:

- If the server process does not have sufficient permissions (that is, if it has not been configured with the correct user ID with sufficient permissions for the applicable database)
- If you used non-default table spaces
- If your servers run Business Process Choregrapher

Related tasks

Upgrading databases for migration
In conjunction with migration, the database schema of some WebSphere Process Server components must be upgraded. This can occur automatically but in some cases you must upgrade the schema manually.

Related reference

WBIPreUpgrade command-line utility
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-line utility
Use the WBIPostUpgrade command for WebSphere Process Server to retrieve the profile configuration that was saved by the WBIPreUpgrade command at the *backupDirectory* that you had specified.

WBIProfileUpgrade script
Use the WBIProfileUpgrade script to update application and configuration settings in a WebSphere Process Server profile when you are migrating clusters and in some other special situations.

Migration wizard

The version-to-version migration wizard is a graphical interface that guides you through migrating from an older version to a newer version of WebSphere Process Server.

Note: The migration wizard cannot run in a non-graphical environment. Examples of non-graphical environments include the **i5/OS platform** or telnet sessions. If you want to run migration in a non-graphical environment, use the WBIPreUpgrade and WBIPostUpgrade commands.

Note: The migration wizard supports only WebSphere Process Server profiles. If you have WebSphere Application Server profiles, you must use the migration commands.

What the migration wizard does

The migration wizard uses the WBIPreUpgrade command and WBIPostUpgrade command to migrate the data and applications from the older version to the newer version of WebSphere Process Server.

This step transfers applications and configuration information for the older version's server resources, security, variables, and virtual hosts to the newer version's server. All stored information is in XML files in the *profile_dir*/config/cells directory of each product.

The WBIPreUpgrade tool saves selected files from the *install_root* and *profile_root* directory to a backup directory you specify on a wizard panel. Migration saves files to the following subdirectories in the backup directory: profiles/*profile_name* and websphere_backup.

Later, the migration wizard uses the WBIPostUpgrade tool to restore a selected profile environment in the backup directory into the newer WebSphere Process Server profile environment

Accessing the migration wizard

Invoke the migration wizard in one of the following ways:

- From the WebSphere Process Server First Steps console, select **Migration wizard**.
- Run one of the following scripts (depending upon your operating system) stored in the *install_dir*/bin directory:
  - Linux UNIX **On Linux and UNIX platforms:** `wbi_migration.sh`
  - Windows **On Windows platforms:** `wbi_migration.bat`

  Note: You can optionally change the default trace setting (`*=all=enabled:com.ibm.ws.migration.common.*=all=disabled`) when invoking the migration wizard. The default trace setting enables tracing on only certain classes, but you can change the default to either enable full tracing or disable all tracing.
  - To enable full tracing, run one of the following scripts to invoke the migration wizard, depending on your operating system:
    - Linux UNIX **On Linux and UNIX platforms:** `wbi_migration.sh -W -migrationPanel.traceString="*=all=enabled"`
    - Windows **On Windows platforms:** `wbi_migration.bat -W -migrationPanel.traceString="*=all=enabled"`
  - To disable all tracing, run one of the following scripts to invoke the migration wizard, depending on your operating system:
    - Linux UNIX **On Linux and UNIX platforms:** `wbi_migration.sh -W -migrationPanel.traceString="*=all=disabled"`
    - Windows **On Windows platforms:** `wbi_migration.bat -W -migrationPanel.traceString="*=all=disabled"`

Before you begin

The migration wizard prompts you for information as you proceed. Before you invoke the wizard, collect the following information:

Cell name
Name of the cell managed by the deployment manager that you are migrating. The cell name for source and target profile for all migration scenarios must match.

Installation root directory
See WBIPreUpgrade command-line utility for a description of the currentWebSphereDirectory parameter.

Migration backup directory name
See WBIPreUpgrade command-line utility for a description of the backupDirectory parameter.

Administrative security user name (required if adminstrative security is configured)
See WBIPostUpgrade command-line utility for a description of the -username parameter.

Administrative security password (required if adminstrative security is configured)
See WBIPostUpgrade command-line utility for a description of the -password parameter.

Source profile name
See WBIPostUpgrade command-line utility for a description of the -oldProfile parameter.

Target profile name
See WBIPostUpgrade command-line utility for a description of the -profileName parameter.

Port value assignments (optional)
See WBIPostUpgrade command-line utility for a description of the -replacePorts and -portBlock parameters.

Note: This applies only if you are migrating from version 6.0.2.x toversion 6.2.

Related reference

WBIPreUpgrade command-line utility
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-line utility
Use the WBIPostUpgrade command for WebSphere Process Server to retrieve the profile configuration that was saved by the WBIPreUpgrade command at the *backupDirectory* that you had specified.

Running the migration wizard:

Run the migration wizard on AIX, HP-UX, Linux, Solaris, or Windows systems to migrate WebSphere Process Server.

Before you begin

Make sure you have performed the required steps that precede the step for invoking the migration wizard. These steps differ depending on whether you are migrating a stand-alone server, a deployment manager, a non-clustered managed node, or a cluster.

About this task

Running the migration wizard is just one step in a larger series of steps required to migrate WebSphere Process Server from version 6.1.x or 6.0.2.x to version 6.2.

Note: If you have a WebSphere Application Server profile that was created in a WebSphere Process Server environment, you must use the WebSphere Process Server command line tools to migrate it. If you have a WebSphere Process Server profile that was created in a WebSphere Process Server environment, you can use either the migration wizard or the command-line tools to migrate it.

Note: The migration wizard cannot run in a non-graphical environment. Examples of non-graphical environments include the **i5/OS platform** or telnet sessions. If you want to run migration in a non-graphical environment, use the WBIPreUpgrade and WBIPostUpgrade commands.

Procedure

1. On the Welcome to the Migration wizard for WebSphere Process Server screen, read the Welcome panel to learn about the migration process, and then click **Next**.

2. On the Detected version of WebSphere Process Server screen, select or specify a previous version of WebSphere Process Server from which to migrate, and then click **Next**.

   Select the check box and enter the location of the previous installation if it does not appear in the selection list.
3. On the Source profile selection screen, select the profile from the previous version of WebSphere Process Server that you want to migrate, and then click **Next**.
4. On the Target profile selection screen, you should select **Create new profile** to create a new, empty target profile for migration. Optionally, you can select a profile from the list of valid profiles for the version 6.2 installation, but it is strongly recommended that you use a new target profile for migration. For more information about creating target profiles for migration, refer to "Target profile considerations" on page 9.
   - If you selected **Create new profile**, enter a name for the profile and leave the host name (name of the system on which the profile will reside) as is, and then click **Next**.

     Note: It is recommended that you do not change the host name but if for any reason you change the host name, then after migration, you will need to manually update the host name in other places where it is configured.
   - You should consider backing up your target profile before you click **Next**. To decide whether to back up your target profile's configuration, consider the following suggestions:
     - If you have selected an already existing profile, you may want a backup of your target profile's configuration. Use the `backupConfig` command to back it up before proceeding.
     - If you are migrating a managed node, the migration will change the deployment manager's configuration. You should use the `backupConfig` command to back up the deployment manager's configuration before proceeding.
5. On the Migration backup directory screen, specify a migration backup directory in which to place a backup copy of the configuration from the previous version. Depending on which version of WebSphere Process Server you are migration from, the following options appear on the screen:
   - **Migrating from version 6.0.2.x**:

     During a migration from version 6.0.2.x to version 6.2, the back up operation backs up all of the profiles created under WebSphere Process Server version 6.0.2.x installation.
     - If this is the first or only profile you are migrating in this WebSphere Process Server installation, keep the **Back up all existing profiles** check box selected. The directory is created if it does not already exist. If the directory exists, it should be empty because the backup operation might overwrite existing backup files.
     - If you have already migrated another profile in this WebSphere Process Server installation and have not made any configuration changes, you can choose not to back up the previous configuration at this point in the migration wizard. In this case, you can clear the **Back up all existing profiles** check box and make sure that the backup directory name is the same as that specified for the previous migration.
   - **Migrating from version 6.1.x**:

During a migration from version 6.1.x to version 6.2, the back up operation backs up only the specified profile intended to be migrated from the WebSphere Process Server version 6.1.x installation

- If the directory does not already exist, it is created during migration.
- If the directory exists, it should be empty because the backup operation might overwrite existing backup files.

After entering the backup directory information, click **Next**.

6. On the Migration settings for Applications screen, specify where the migrated applications should be located and then click **Next**.

   Note: The Migration settings for Applications screen appears only if you are migrating from version 6.0.2.x to version 6.2.

   You can choose any one of the following options:

   - Keep the current application installation directories.

     Restrictions: If you choose this option, the location is shared by the existing installation and the new installation. If you keep the migrated applications in the same locations as those of the previous version, the following restrictions apply:

     - Mixed-node support limitations must be followed. This means that the following support cannot be used when evoking the wsadmin command:
       - Precompile JSP
       - Use Binary Configuration
       - Deploy EJB
     - You risk losing the migrated applications unintentionally if you later delete applications from these locations when administering (uninstalling for example) your version 6.0.2.x installation.
   - Install the applications in the default directory of the target installation.
   - Install the applications in the following directory.

     In the directory field, you can either type the directory path or click **Browse** to search for it. It is recommended that you use the following directory path: `${USER_INSTALL_ROOT}\installedApps`.

7. On the Deployment manager option screen, you can optionally select **Do not disable the deployment manager of the previous version**. It is recommended that you leave this option unchecked to prevent conflicts with the version 6.2 deployment manager.

   Note: The Deployment manager option screen appears only if you are migrating from version 6.0.2.x to version 6.2.

8. On the Port value assignment screen, select one of the options for assigning port values, and then click **Next**.

   Note: The Port value assignment screen appears only if you are migrating from version 6.0.2.x to version 6.2.

   You can choose to do any one of the following with the port values:

   - Use the port values assigned to the previous (source) installation.
   - Use the port values that are assigned to the specified target profile.
   - Define a block of port values starting at:

     If you select this option, enter the first value of the block of consecutive port numbers to assign.

9. On the Additional migration options screen, enter the following information, then click **Next**.

   Important: The Support 6.0.x scripts in target profile check box is visible only if you are migrating from version 6.0.2.x.

   - Support 6.0.x scripts in target profile check box: If you are migrating from version 6.0.2.x, you can select this check box if you want to migrate to support script compatibility. If you select this option, the migration wizard creates the following configuration definitions of the version 6.0.2.x of WebSphere Process Server:
     - Transport
     - ProcessDef
     - 6.0.2 SSL
     - 6.0.2 ORB service threadpool

     instead of the following version 6.2 configuration definitions:
     - Channels
     - ProcessDefs
     - version 6.2 SSL
     - version 6.2 ORB service threadpool

     Select this option in order to minimize impacts to existing administration scripts. If you have existing **wsadmin** scripts or programs that use third-party configuration APIs to create or modify your existing configuration definitions, for example, you might want to select this option during migration.

     Note: This is temporary until all of the nodes in the environment are at the newer level (for example, WebSphere Process Server version 6.2.) When they are all at the new level, you should perform the following actions:

     a. Modify your administration scripts to use all of the settings for the new version (in this case version 6.2).
     b. Use the convertScriptCompatibility command to convert your configurations to match all of the settings corresponding to the new version.

        See the convertScriptCompatibility command.

        Note: When following the directions at this link to use the convertScriptCompatibility command, use the WBIPostUpgrade command rather than the WASPostUpgrade command.
   - Enter security credentials check box: If security was not enabled on the prior version of WebSphere Process Server make sure the **Enter security credentials** check box is cleared and select **Next**.

     If administrative security was enabled on the version of WebSphere Process Server you are migrating from, the **Enter security credentials** check box must be selected and a corresponding user name and password must be entered in the user name and password fields.

     If the migration wizard detects the user name and password that was configured in the earlier version of WebSphere Process Server, then the **Enter security credentials** check box will be automatically selected and the correct user name and password will be automatically configured in the user name and password fields. If this occurs, verify that the user name entered is correct (the password will be masked) and select **Next**.

If security was enabled in the prior version, but the migration wizard cannot detect the user name and password, it will check the **Enter security credentials** check box but leave the user name and password fields blank. If this occurs, enter the user name and password (the same as those used for administrative security in the previous version) and select **Next**.

Note: Whether or not the wizard can detect the user name and password depends on which version of WebSphere Process Server you are migrating from.

10. On the Migration summary screen, check the information in the summary panel and make sure that it is correct, and then click **Next** to start the migration. During migration, the following panels may appear:
    - If you selected the option to create a new target profile, screens show the beginning and results of that creation.
    - Screens show the progress of the migration process.
    - If the migration is not successful, the wizard displays a failure screen. If the migration is partially successful, the wizard displays a warning screen. Correct any problems, and try the migration again.
    - If the post-migration is successful, the wizard displays an indication of success.
11. Click **Finish** to exit the migration wizard.

Results

You can now start the migrated server in the WebSphere Process Server environment at the new release level.

Related concepts

Migration wizard
The version-to-version migration wizard is a graphical interface that guides you through migrating from an older version to a newer version of WebSphere Process Server.

Related tasks

Creating profiles
Learn how to create new WebSphere Enterprise Service Bus or WebSphere Process Server profiles. You can create profiles from a command line by using the manageprofiles command, or interactively by using the Profile Management Tool graphical user interface (GUI).

Related reference

WBIPreUpgrade command-line utility
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-line utility
Use the WBIPostUpgrade command for WebSphere Process Server to retrieve the profile configuration that was saved by the WBIPreUpgrade command at the *backupDirectory* that you had specified.

Related information

Java virtual machine settings

convertScriptCompatibility command

How data is handled during migration from earlier versions

The WebSphere Process Server version-to-version migration tools will handle different sets of data (application data, configuration data, database information, and long-running processes) in different ways.

- "Application data"
- "Configuration data" on page 27
- "Database information" on page 27
- "Long-running processes" on page 28

Application data

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 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 6.2 information center.

Note: If you have SCA modules which use a single reference for both dynamic and static invocations, and the reference is wired to an import with a JMS or HTTP binding, then the JMS or HTTP binding will now be used for dynamic invocations using jms: or http: URLs, rather than performing a dynamic web service invocation. To retain the version 6.1.2 behavior and continue to make Web service invocations in this scenario, you must either update your module to correctly set the bindingType to indicate a web service URL when making the invocation (for MFC or POJO components) or otherwise set the WebSphere variable SCA_USE_WS_FOR_DYNAMIC_INVOCATION to include the name of the modules in a semi-colon delimited list, e.g. sca/myModule1;sca/myModule2

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.

Sample applications are handled differently. For stand-alone profiles, the migration process will not install any sample applications. To make sample applications available for a stand-alone profile, you can install them using the installation wizard for the later version of WebSphere Process Server. For network deployment profiles, any samples installed with the previous version of WebSphere Process Server will be installed during migration to the new version.

Configuration data

The version-to-version migration tools (wizard or 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.

Database information

Automatic database migration

If you are migrating from version 6.0.2.x and have a Cloudscape database, the migration tools will migrate the database configuration automatically, with certain exceptions. See “Migrating Cloudscape databases” on page 128 for more information. In addition, the Cloudscape database will be converted to a Derby database, which is the successor to Cloudscape and is supported by WebSphere Process Server version 6.2.

Manual database migration

If you have a database other than Cloudscape, the migration tools will automatically migrate the provider and datasource definitions for each existing data source and provider. However, database schema upgrades may also be required, which could require special attention. If the server process has the required database permissions and, in the case of some databases, meets other requirements, the schema upgrades will happen automatically when the server is first started.

If the server process does not have the required permissions or meet other requirements, or if you would like to manually upgrade your database schemas, then you will need to use the scripts provided.

If you have Business Process Choreographer or Business Space configured, you must update the database manually.

See "Upgrading databases for migration" on page 34 for more information.

Long-running processes

Long-running business process instances and human task instances are taken care of during version-to-version migration as the databases (storing the instances) are taken over. During migration, the database schema is upgraded and the data is converted to the new schema. After migration, those instances continue to run in the migrated environment.

Related concepts

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

Related tasks

Migrating Cloudscape databases
After you use the migration tools to migrate to WebSphere Process Server version 6.2, you should verify the results of the automatic Cloudscape database migration and manually migrate any Cloudscape database instances that are not automatically migrated by the tools.

Configuration mapping during product-configuration migration

Various configurations are mapped during product-configuration migration.

Migration always involves migrating a single profile to another single profile on the same system or a separate system. Examples include a WebSphere Process Server version 6.1 deployment manager migrating to a version 6.2 deployment manager profile and a version 6.1 stand-alone server migrating to a version 6.2 stand-alone server profile.

Note: Only a stand-alone server profile can be migrated to a separate system.

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.

Bootstrap port

The migration tools map a non-default value directly into the version 6.2 environment. When migratng from version 6.0.2.x, if the `-portBlock` parameter is specified during the call to WBIPostUpgrade, a new port value is given to each server that is migrated to version 6.2.

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 6.2 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 Network Deployment, version 6.1 information center. For information on how to change the Java virtual machine settings, see Java virtual machine settings in the WebSphere Application Server Network Deployment, version 6.1 information center.

Java heap size for migrating EAR files

When migrating all WebSphere Process Server EAR files to version 6.2 using the wsadmin tool, the WBIPostUpgrade tool uses the default maximum Java heap size value of 64 MB to install the EAR files.

If an EAR file fails to install during migration because the Java heap size is not large enough, you will see a message similar to the following message:

```
java.lang.OutOfMemoryError JVMXE006:OutOfMemoryError
```

Increase the maximum Java heap size and follow the example below to install the application.

Example of installing an application on WebSphere Process Server version 6.2

Assume that:

Installation root
C:\WebSphere\DeploymentManager

Number signs (###)
Maximum heap size value

<EAR_file_name>
Name of the EAR file

app_name
Name of the application

cluster_name
Name of the cluster on which the EAR file should be installed

The command is displayed on more than one line for clarity.

```
wsadmin -conntype NONE
        -javaoption
        -Xmx###m
        -c "$AdminApp install
            C:\\WebSphere\\ProcServer
                   <EAR_file_name>
        {-nodeployejb
         -appname app_name
         -cluster cluster_name}"
```

Migration of a version 6.1.x or 6.0.2.x node to a version 6.2 node

You can migrate a WebSphere Process Server version 6.1.x or 6.0.2.x node that belongs to a cell to WebSphere Process Server version 6.2 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 6.1.x or 6.0.2.x to version 6.2. If you use a different cell name, federated nodes cannot successfully migrate to the WebSphere Process Server version 6.2 cell.

Migrating a base WebSphere Process Server node that is within a cell to version 6.2 also migrates the node agent to version 6.2.

A cell can have mixed nodes, which means it can contain some version 6.2 nodes and some version 6.1.x nodes.

Note: Mixed nodes are not supported if you are migrating from version 6.0.2.x.

Policy file

WebSphere Process Server version 6.2 migrates all the policy files that are installed with version 6.1.x or 6.0.2.x policy files with the following characteristics:

- Any comments located in the version 6.2 policy file will be preserved. Any comments contained in theversion 6.1.x or 6.0.2.x policy file will not be included in the version 6.2.
- 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 6.2 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 `MIGR0372I: 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 6.2 configuration.

Property files

WebSphere migrates all the property files that are installed with version 6.1.x or 6.0.2.x by merging settings into the version 6.2 property files.

Migration does not overlay property files.

Resource adapter archives (RARs) referenced by J2C resources

RARs and JARs that are referenced by J2C resources are migrated as follows:

Migrating cluster-level resources

Cluster-level resources are configured in resource*xxx*.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 resource*xxx*.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 6.1.x or 6.0.2.x to version 6.2:

- **Migration from version 6.0.2.x to version 6.2**: The migration copies files such as RAR or JAR files from WAS_INSTALL_ROOT to WAS_INSTALL_ROOT and from USER_INSTALL_ROOT to USER_INSTALL_ROOT
- **Migration from version 6.1.x to version 6.2**: The migration copies configuration files as follows:
  - If you install RAR or JAR as part of WebSphere Process Server installation, then the configuration files are migrated to the migration target profile and updated to point to the new version of the RAR and JAR files.
  - If you install RAR or JAR files after the installation of WebSphere Process Server, then the following will occur
    - If you install the RAR or JAR files under the previous WebSphere Process Server installation, only the configuration files are migrated, and you need to either copy or install those RAR or JAR files on the migration target profile and make sure the configuration is correct before starting the server.
    - If you install the RAR or JAR files outside of the previous WebSphere Process Server installation (which is recommended), then the configuration files are migrated and you do not need to take any action after migration.

If you hardcoded a path to a RAR file (archivePath="C:/WAS/installedConnectors/x2.rar" for example) in version 6.1.x or 6.0.2.x, however, the version 6.2 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 migration of a stand-alone profile, no WebSphere Process Server samples are migrated. Equivalent version 6.2 samples are available for all version 6.2 samples

Security

Note: The following security information applies only if you are migrating from version 6.0.2.x

Java 2 security is enabled by default when you enable security in WebSphere Process Server version 6.2. 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 6.2. 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 6.2 properties directory to enterprise applications as they are being migrated.

When migrating from WebSphere Process Server version 6.0.2.x to version 6.2, 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 6.2 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 Serverversion 6.2. The default version 6.2 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.

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 6.0.2.x. For more information on migrating your security configurations to version 6.2, see Migrating, coexisting, and interoperating - Security considerations in the WebSphere Application Server Network Deployment, version 6.1 information center.

Stdin, stdout, stderr, passivation, and working directories

The location for these directories is typically within the installation directory of a previous version. The default location for stdin, stdout, and stderr is the logs directory of the WebSphere Process Server version 6.2 installation root.

The migration tools attempt to migrate existing passivation and working directories. Otherwise, appropriate version 6.2 defaults are used.

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

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 command, a new value is assigned to each transport that is migrated.

For more information on the WBIPostUpgrade command, see WBIPostUpgrade command-line utility.

For further information on transport chains and channels, see Transport chains.

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 6.1.x or 6.0.2.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 for WebSphere Process Server
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 Network Deployment Information Center.

Migrating clusters
Migrate clusters by migrating, in turn, each profile that contains cluster members following special procedures. Take additional steps if you want to minimize down time of cluster services.

Related tasks

Migrating non-clustered managed nodes
Migrate a WebSphere Process Server managed node, choosing from several methods depending on your needs.

Related reference

WBIPostUpgrade command-line utility
Use the WBIPostUpgrade command for WebSphere Process Server to retrieve the profile configuration that was saved by the WBIPreUpgrade command at the *backupDirectory* that you had specified.

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 Network Deployment 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 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 6.2 information center.

For more information about migrating certain types of WebSphere applications, see Migrating WebSphere Applications and its subsidiary topics in the WebSphere Application Server Network Deployment, version 6.1 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 6.2 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
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.

Upgrading databases for migration

In conjunction with migration, the database schema of some WebSphere Process Server components must be upgraded. This can occur automatically but in some cases you must upgrade the schema manually.

About this task

Normally, database changes required by new versions of WebSphere Process Server are made automatically. When the server is first started the database tables are migrated to the new schema version. However, if you have Business Process Choreographer or Business Space configured, or if the server has insufficient permissions to access the database schema or other database-specific requirements are not met, you must update the database manually.

For WebSphere Process Server version 6.2, the databases requiring schema upgrades are the following:

- Common database (default name WPRCSDB)
- Business Process Choreographer (default name BPEDB)
- Business Space powered by WebSphere (default name IBMBUSSP)

Perform the manual schema upgrade after migration, but before starting any server that uses the database.

For instructions on manually upgrading these databases, see the following subtopics.

Related tasks

Migrating Cloudscape databases
After you use the migration tools to migrate to WebSphere Process Server version 6.2, you should verify the results of the automatic Cloudscape database migration and manually migrate any Cloudscape database instances that are not automatically migrated by the tools.

Related reference

WBIPreUpgrade command-line utility
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-line utility
Use the WBIPostUpgrade command for WebSphere Process Server to retrieve the profile configuration that was saved by the WBIPreUpgrade command at the *backupDirectory* that you had specified.

WBIProfileUpgrade script
Use the WBIProfileUpgrade script to update application and configuration settings in a WebSphere Process Server profile when you are migrating clusters and in some other special situations.

Upgrading the Common database manually

After migrating the server from a previous version, you must upgrade to a new database schema for the ″Common″ database before you start the server. You must upgrade manually if the database user that is defined for the data source does not have sufficient authorization to modify the database schema.

Before you begin

- You should have already run the migration wizard or migration scripts to migrate the server, or in the case of a cluster, the servers in the cluster.
- Make sure the server or, if applicable, servers in the cluster remain stopped (do not start them after the migration wizard or scripts have run before completing the database upgrade).

About this task

Any database that is accessed by a migrated server needs to have its schema updated before you start the server. In the case of a cluster, any database that is accessed by any of the migrated cluster members needs to have its schema updated before you start any of the cluster members. You must upgrade manually if the database user that is defined for the data source does not have sufficient authorization to modify the database schema. For the Common database, the database user that is configured for the data source must be authorized to perform all of the following operations: create and alter tables and create and drop indexes and views.

Procedure

1. Make sure that you are using a user ID with sufficient authority to update the database schema.
2. Back up the database.
3. Locate the directory where the database scripts are located:
   - i5/OS Linux UNIX On Linux, UNIX, and i5/OS platforms: *install_root*/dbscripts/*component_name*/*database_type*
   - Windows On Windows platforms: *install_root*\dbscripts*component_name**database_type*

   Where:

   install_root
   is the root directory into which WebSphere Process Server version 6.2 has been installed.

   component_name
   is CommonDB.

   database_type
   is a name corresponding to the database product you are using. Applicable database types and their directory names are as follows:

Database type	Directory name
DB2 Universal Database (for all operating systems except z/OS and i5/OS)	DB2
DB2 Universal Database for i5/OS	DB2iSeries
DB2 for z/OS Version 8.x	DB2zOSV8 - Use scripts in this directory if your initial database configuration used DB2 z/OS v8 (uses long table names) or if you upgraded from DB2 z/OS v7 to DB2 z/OS v8
DB2 for z/OS Version 9.x	DB2zOSV9 - Use scripts in this directory if your initial database configuration used DB2 z/OS v9 or later (uses long table names) or if you upgraded from DB2 z/OS v7 to DB2 z/OS v9.
Derby	Derby **Note:** If you have an existing Cloudscape database for a previous (6.0.2) installation of WebSphere Process Server, the database must first be upgraded from a Cloudscape database to a Derby database. See “Migrating Cloudscape databases” on page 128. When you have a Derby database that corresponds to that same (6.0.2) WebSphere Process Server version, you must upgrade the database to correspond to WebSphere Process Server version 6.2 with the scripts in this directory.
Informix®	Informix
Oracle	Oracle
Microsoft SQL Server	SQLServer

   For specific database product names and versions see Database configurations.

4. Locate the migration scripts for the database that starts with word **upgradeSchema**. Based on your database(s), you will find upgradeSchema.bat, upgradeSchema.sh, or both, as well as component-specific SQL scripts like `upgradeSchema<Migrated WBI version>_<component>.sql`. For example, to upgrade the common database schema from WebSphere Process Server version 6.0.2.x to version 6.2 and `dbType` is `DB2_UNIVERSAL`, you need the following migration scripts:
   - `install_root/dbscripts/CommonDB/DB2/upgradeSchema.bat`
   - `install_root/dbscripts/CommonDB/DB2/upgradeSchema.sh`
   - `install_root/dbscripts/CommonDB/DB2/upgradeSchemaTables.bat`
   - `install_root/dbscripts/CommonDB/DB2/upgradeSchema602_Recovery.sql`
   - `install_root/dbscripts/CommonDB/DB2/upgradeSchema602_relationshipService.sql`
   - `install_root/dbscripts/CommonDB/DB2/upgradeSchema602_governancerepository.sql`
   - `install_root/dbscripts/CommonDB/DB2/upgradeSchema602_DirectDeploy.sql`
   - `install_root/dbscripts/CommonDB/DB2/upgradeSchema602_CommonDB.sql`
   - `install_root/dbscripts/CommonDB/DB2/upgradeSchema602_customization.sql`

   Note: Other scripts that exist in the same CommonDB/DB2 directory may be used, depending on what version of WebSphere Process Server you migrated from.
5. Copy the appropriate script or scripts from the directory in which you found it to the system on which the database is hosted.
6. Check the SQL scripts, and modify them, if necessary, to meet your requirements. This includes any scripts in the directory that are named according to the following syntax: upgradeSchema*nnn*_<component>xql where *nnn* corresponds to a product version number and "component" corresponds to any one of the following:
   - CommonDB
   - relationshipService
   - governancerepository
   - DirectDeploy
   - customization

   For example, you might need to change a user name, password, or file path.
7. Using your database client, connect to the database. This is to ensure that it can connect.
8. Run your version of the upgrade SQL scripts. For information on how to execute a .sql script with your database, refer to the documentation for your database product.

   If there are any errors, or failure is indicated in your database client output, fix the reported errors and try this step again.

Results

The database schema has been updated. When the server is started for the first time after the upgrade, the data is migrated according to the new schema. After the data has been migrated,version 6.1.x or version 6.0.2.x servers cannot run against the database.

Related tasks

Upgrading the Business Process Choreographer database manually
After migrating a server or cluster that has Business Process Choreographer configured, you must manually upgrade the schema for the Business Process Choreographer database and perform a data migration before you start the server or any cluster member.

Upgrading the Business Process Choreographer database manually

After migrating a server or cluster that has Business Process Choreographer configured, you must manually upgrade the schema for the Business Process Choreographer database and perform a data migration before you start the server or any cluster member.

Before you begin

- You should have already run the migration wizard or migration scripts to migrate the server, or in the case of a cluster, the servers in the cluster.
- Make sure the server or, if applicable, servers in the cluster remain stopped (do not start them after the migration wizard or scripts have run before completing the database upgrade).

About this task

For production systems, the database associated with Business Process Choreographer that is accessed by a migrated server needs to have its schema updated before you start the server. In the case of a cluster, any such database that is accessed by the migrated cluster members needs to have its schema updated before you start any of the cluster members.

Procedure

1. If you are using DB2 for z/OS and OS/390® Version 7, and have not yet upgraded the database to DB2 for z/OS Version 8 or DB2 9 for z/OS, perform the upgrade now, as described in the DB2 for z/OS documentation.
2. Back up the Business Process Choreographer database according to the documentation for your database.
3. Upgrade the database schema as described in “Upgrading the Business Process Choreographer database schema” on page 39
4. Migrate the runtime data in your database as described in “Migrating the Business Process Choreographer runtime data” on page 42.
5. Start the server or cluster to verify that the Business Process Choreographer database migration was successful. If you cannot get the server to start, you might have to restore your database from the backup, and repeat from 3.

Results

The database schema has been updated and the data has been migrated to the new schema. This means that version 6.1.x or version 6.0.2.x servers cannot run against the database.

Related concepts

Premigration considerations for Business Process Choreographer
If your servers run Business Process Choreographer, you should be aware of certain things that you need to plan and take into consideration before you migrate Business Process Choreographer.

Related tasks

Upgrading the Common database manually
After migrating the server from a previous version, you must upgrade to a new database schema for the ″Common″ database before you start the server. You must upgrade manually if the database user that is defined for the data source does not have sufficient authorization to modify the database schema.

Migrating non-clustered managed nodes using the migration wizard
Migrate non-clustered managed nodes from an older version to a newer version of WebSphere Process Server using the migration wizard.

Migrating non-clustered managed nodes using the command-line tools
Migrate non-clustered managed nodes from an older version to a newer version of WebSphere Process Server with the command-line tools.

Migrating a cluster
To migrate a cluster, migrate each profile containing a member of that cluster one at a time. The migration requires extra steps not required for a non-clustered environment.

Migrating a cluster with minimal down time
To migrate a cluster while minimizing down time, first migrate approximately half of the profiles contributing to the cluster, then migrate the other half. Perform the extra steps required for cluster migration after you have migrated the first set of profiles.

Upgrading the Business Process Choreographer database schema

After migrating or upgrading a server or cluster that has Business Process Choreographer configured, the schema for the associated Business Process Choreographer database must be upgraded.

Procedure

1. Locate the directory where the database scripts are located:
   - i5/OS Linux UNIX On Linux, UNIX, and i5/OS platforms: *install_root*/dbscripts/*component_name*/*database_type*
   - Windows On Windows platforms: *install_root*\dbscripts*component_name**database_type*

   Where:

   install_root
      is the root directory into which WebSphere Process Server version 6.2 has been installed.

   component_name
      Is ProcessChoreographer.

   database_type
      is a name corresponding to the database product you are using. Applicable database types and their directory names are as follows:

Database type	**Directory name**
DB2 Universal Database (for all operating systems except z/OS and i5/OS)	DB2
DB2 Universal Database for i5/OS	DB2iSeries
DB2 for z/OS Version 8.x	DB2zOSV8 - Use scripts in this directory if your initial database configuration used DB2 z/OS v8 (uses long table names) or if you upgraded from DB2 z/OS v7 to DB2 z/OS v8
DB2 for z/OS Version 9.x	DB2zOSV9 - Use scripts in this directory if your initial database configuration used DB2 z/OS v9 or later (uses long table names) or if you upgraded from DB2 z/OS v7 to DB2 z/OS v9.
Derby	Derby **Note:** If you have an existing Cloudscape database for a previous (6.0.2) installation of WebSphere Process Server, the database must first be upgraded from a Cloudscape database to a Derby database. See "Migrating Cloudscape databases" on page 128. When you have a Derby database that corresponds to that same (6.0.2) WebSphere Process Server version, you must upgrade the database to correspond to WebSphere Process Server version 6.2 with the scripts in this directory.
Informix	Informix
Oracle	Oracle
Microsoft SQL Server	SQLServer

For specific database product names and versions see Database configurations.

2. Prepare to copy the migration scripts that you need for your database and current schema version, where *schema_version* has the value `602` for version 6.0.2, `610` for version 6.1.0, or `612` for version 6.12. Locate the scripts that you need, but do not run them yet.

   For DB2 on Linux, UNIX, and Windows:

   Use upgradeTablespaces*schema_version*.sql to upgrade the table spaces prior to upgrading the database objects, and one of the following upgrade scripts:

   - upgradeSchema*schema_version*.sql to create new database objects in the table spaces that were created using the createTablespace.sql script at schema creation time.
   - upgradeSchema*schema_version*nonp.sql to create new objects in the default table space.

   For DB2 on i5/OS:

   Use the script upgradeSchema*schema_version*.sql to upgrade the schema objects.

   Note: There is no upgradeTablespaces*schema_version*.sql script for DB2 on i5/OS.

   For DB2 on z/OS and OS/390:

- If you used DB2 z/OS V8 or V9 during your initial database configuration, use both of the following scripts, in the following order:
  a. upgradeTablespaces*schema_version*.sql to upgrade the table spaces prior to upgrading the database objects.
  b. upgradeSchema*schema_version*.sql to upgrade the database objects after you upgraded the table spaces.
- If you used DB2 z/OS V7 during your initial database configuration, use both of the following scripts, in the following order:
  a. upgradeTablespaces*schema_version*DB2zOSV7.sql to upgrade the table spaces prior to upgrading the database objects.
  b. upgradeSchema*schema_version*DB2zOSV7.sql to upgrade the database objects after you upgraded the table spaces.

For Derby:

Use one of the following upgrade scripts:

- upgradeSchema*schema_version*.sql to upgrade the schema using a schema qualifier.
- upgradeSchema*schema_version*nonp.sql to upgrade the schema without using a custom schema qualifier.

For Informix Dynamic Server:

Use one of the following upgrade scripts:

- upgradeSchema*schema_version*.sql to create new database objects in the database spaces that were created using the createDbspace.sh or createDbspace.bat shell script at schema creation time.
- upgradeSchema*schema_version*nonp.sql to create new objects in the default database space.

For Oracle

Use the script upgradeSchema*schema_version*.sql.

Note: There is no upgradeSchema*schema_version*nonp.sql script to create new objects in the default table space. In this case, only the automatic schema upgrade is supported.

For Microsoft SQL Server

Use one of the following upgrade scripts:

- upgradeSchema*schema_version*.sql to upgrade the schema using a custom schema qualifier.
- upgradeSchema*schema_version*nonp.sql to upgrade the schema without using a custom schema qualifier.
- upgradeSchema*schema_version*Unicode.sql if you have created the schema with UNICODE support using the createSchemaUnicode.sql script or createDatabaseUnicode.sql script and if you want to use a custom schema qualifier.
- upgradeSchema*schema_version*UnicodeNonp.sql if you have created the schema with UNICODE support using the createSchemaUnicode.sql script or createDatabaseUnicode.sql script and if you do not want to use a custom schema qualifier.

Note: The upgradeSchema...Nonp.sql versions of the Microsoft SQL Server upgrade scripts do not make use of a schema qualifier. They upgrade the database objects in the user schema.

3. Copy the appropriate script or scripts from the directory in which you found it to the system on which the database is hosted.
4. Check the SQL script or scripts you just copied, and modify them, if necessary, to meet your requirements. Depending on the script and the database, you might need to change a user name, password, schema qualifier, or file path. For example, if you are using DB2 Universal Database for i5/OS, you must replace all occurrences of the @SCHEMA@ placeholder in the SQL statements with your Business Process Choreographer database collection name.
5. Using your database client, connect to the database. This is to ensure that it can connect.
6. If you are using DB2 Universal Database for i5/OS, set up the IBM System i® environment to automatically reply to any inquiry messages sent when running the ALTER table commands (inquiry messages typically require an interactive user response).
   a. Open an i5/OS command line window.
   b. Enter `DSPJOB`, select option `2` **Display job definition attributes** and record the original value for **Inquiry message reply** .
   c. Then enter the following commands:

      ```
      CHGJOB INQMSGRPY(*SYSRPYL)
      ADDRPYLE SEQNBR(nn) MSGID(CPA32B2) CMPDTA(*NONE) RPY(I)
      ```

      Where *nn* is an unused sequence number in the system reply list.
   d. Start a QShell session.
7. Run your version of the upgrade SQL scripts. For information on how to execute a .sql script with your database, refer to the documentation for your database product. For example, if you are migrating from version 6.1.0, using DB2 Universal Database for i5/OS issue the following command in QShell: `db2 -tvf upgradeSchema610.sql`

   If there are any errors, or failure is indicated in your database client output, fix the reported errors and retry this step.
8. If you are using DB2 Universal Database for i5/OS, restore the original ″Inquiry message reply″ value
   a. In an i5/OS command line window, enter the command to list the reply list entries:

      ```
      WRKRPLYLE
      ```
   b. Select the reply that was added in step 6c, and enter option 4 (Delete) next to that entry.
   c. Then enter the following command:

      ```
      CHGJOB INQMSGRPY(original_value)
      ```

Results

The Business Process Choreographer database schema has been updated.

What to do next

Perform the Business Process Choreographer data migration.

Migrating the Business Process Choreographer runtime data

After migrating or upgrading a server or cluster that has Business Process Choreographer configured, you must perform a data migration before you start the server or any cluster member.

Before you begin

Refer to the following Technote for the most up-to-date information about performing the data migration: Technote 21327385.

Procedure

1. If you are using DB2 for Linux, UNIX, Windows, or z/OS, drop any custom created indexes, views, triggers, and that reference any of the following the tables that the data migration affects:
   - PROCESS_TEMPLATE_B_T
   - ACTIVITY_TEMPLATE_B_T
   - SCOPED_VARIABLE_INSTANCE_B_T
   - CORRELATION_SET_INSTANCE_B_T
   - STAFF_QUERY_INSTANCE_T
   - TASK_TEMPLATE_T
   - TASK_INSTANCE_T
2. Run the database migration script as described in "Business Process Choreographer data migration script" on page 44.

   Important: Depending on the quantity of data and the power of your database server, the data migration process can take several hours. If the migration fails, there is an option that allows you to restart it and it will continue from where it stopped. Otherwise if it cannot continue or if you stop it because it is taking too long, restore your database from the backup.
3. Verify that the data migration is progressing correctly. The following messages are written to the wsadmin trace file, however, because all the tables are migrated in parallel, the messages for the different tables can be interleaved:
   a. If the data migration does not need to be performed:
      ```
      INFO: CWWBB0642I: No data migration needed for the given database.
      Data Migration finished without any actions.
      ```
   b. If there are any custom tables, you will get the following message:
      ```
      Warning: Custom tables have been configured. They must be dropped and re-created now.
      ```

      You must drop the custom tables, then restart the script.
   c. If another instance of the migration script is already running, you will get the following message:
      ```
      CWWBB0654E: The data migration has already been started.
      ```

      This mechanism is to prevent multiple instances of the migration script running at the same time. If you are sure that all previous attempts to run the script have resulted in error messages, are no longer running, and the problems have been fixed, it is possible to use the `-force` option to by-pass this protection mechanism. For more information about using this option, refer to "Business Process Choreographer data migration script" on page 44.
   d. When the data migration starts:
      ```
      INFO: CWWBB0650I: Start of data migration.
      ```
   e. The start and end of the work item data migration are indicated by:
      ```
      INFO: CWWBB0644I: Start of workitem migration.
      INFO: CWWBB0645I: Workitem migration successfully completed.
      ```

      During the work item data migration, the percentage progress is written approximately every two minutes, for example:
      ```
      Nov 13, 2008 5:04:50 PM  INFO: CWWBB0656I: 'Workitem migration 23.56%' completed.
      ```

f. If your database requires a table space migration, the start and end are indicated by:

```
INFO: CWWBB0646I: Start of tablespace migration.
INFO: CWWBB0647I: Tablespace migration successfully completed.
```

During the table space migration, the start of each table space migration is indicated by a message similar to the following:

```
INFO: CWWBB0657I: Migrating table '1/7'.
```

During the table space migration, the percentage progress is written every two minutes, for example:

```
INFO: CWWBB0656I: 'Table 1/7 95.8%' completed.
```

Completion is indicated by a message similar to the following:

```
INFO: CWWBB0656I: 'Table 1/7 100.0%' completed.
```

g. If an error occurs that prevents the data migration from completing successfully:

```
SEVERE: CWWBB0652E: Data migration finished with an error.
```

In this case check any available stack trace and correct the cause of the problem. After you have corrected the problem, run the data migration script again, as described in step 2 on page 43. The script will try to continue from where it stopped.

Note: You cannot start the Business Flow Manager or Human Task Manager until all of the data has been successfully migrated, and any attempt to start a server that has a Business Process Choreographer configuration on it will result in the following message being written to the SystemOut.log file:

```
SEVERE: CWWBB0653E: The data migration has been started but not finished yet.
```

h. When all of the data has been successfully migrated, it is indicated by:

```
INFO: CWWBB0651I: Data migration finished successfully.
```

i. If custom tables or named materialized views are registered, a warning is displayed after the migration finishes. Materialized views are dropped and re-created automatically, but you must drop and re-create any custom tables manually.

4. After the migration is complete, if you are using DB2 for Linux, UNIX, Windows, or z/OS, re-create any custom objects that you deleted in step 1 on page 43.

Results

The Business Process Choreographer runtime data has been migrated to the new schema. If you have a DB2 database it now uses larger tables space pages.

Business Process Choreographer data migration script:

Use the migrateDB.py script to migrate the runtime data in the Business Process Choreographer database to the new schema, which results in better query performance for business processes and human tasks.

Purpose

This script file merges two tables that contain information about work items. For DB2 databases on Linux, UNIX, Windows, and z/OS, it also performs a migration to tables spaces with a larger (8 k) page size. The goal of this one-time data migration is to improve performance.

Location

This script is located in the Business Process Choreographer subdirectory for administration scripts:

On Linux, UNIX, and i5/OS platforms:

```
install_root/ProcessChoreographer/admin/migrateDB.py
```

On Windows platforms:

```
install_root\ProcessChoreographer\admin\migrateDB.py
```

Restrictions

- This data migration is mandatory. It must be completed before you restart any version 6.2 servers or clusters that have Business Process Choreographer configured on them.
- If this script fails, there is no rollback possibility, so you must backup your database before running the script. However, if the script is restarted, it will attempt to continue migrating the data.
- This script must be run in disconnected mode, which means that the server or cluster that hosts the Business Process Choreographer configuration must be stopped for the duration of the migration.
- If you specify a cluster, you must run the script on the node of a cluster member, and not on the deployment manager.
- The script uses a semaphore to prevent multiple instances of the script running at the same time. If the script is restarted before the first instance has completed, the second instance fails with a com.ibm.bpe.api.DatabaseDataMigrationAlreadyRunningException exception and `CWWBB0654E: The data migration has already been started.` message.
- Depending on the quantity of data and the power of your database server, the migration process can take several hours. For more detailed information about the data migration, refer to Technote 21327385.
- On i5/OS, you must run the script using qshell.

Invoking the script

If you specify a server, you must run the script on the server node. If you specify a cluster, you must run the script on the node of a cluster member, and not on the deployment manager. Because wsadmin overwrites its trace file, use the `-tracefile` option to specify a file name and location for the trace file for the data migration.

To invoke the script, using the profile “profile”, and sending the trace information to the file *migration_trace_file*, enter one of the following commands.

On Linux and UNIX platforms:

```
wsadmin.sh -conntype NONE -profileName profile
       -tracefile migration_trace_file
       -f install_root/ProcessChoreographer/admin/migrateDB.py
       parameters
```

On i5/OS platforms:

```
wsadmin -conntype NONE -profileName profile
        -tracefile migration_trace_file
        -f install_root/ProcessChoreographer/admin/migrateDB.py
        parameters
```

On Windows platforms:

```
wsadmin -conntype NONE -profileName profile
        -tracefile migration_trace_file
        -f install_root\ProcessChoreographer\admin\migrateDB.py
        parameters
```

Parameters

The script takes the following parameters:

```
(-server serverName) | (-cluster clusterName)
[[-dbUser userID] -dbPassword password]
[-dbSchema schema]
[-slice slice]
[-force]
```

Where:

-server *serverName* **|** **-cluster** *clusterName*
Specifies the Business Process Choreographer configuration, for which the database will be migrated.

-dbUser *userID*
The optional user ID to authenticate with the database. If no -dbUser is provided, the default is the user from the data source's authentication alias, and if no authentication alias is set, then no user qualifier is used to connect to the database. If the default user ID has insufficient permissions for the database migration actions, you must use this option to specify a user ID that does have the necessary permissions. For example, for DB2, if you specify -dbUser, and are using an implicit schema based on a different user ID, then you must specify the -dbSchema parameter.

-dbPassword *password*
The password is required, except if you use a Derby database.

-dbSchema *schema*
This parameter is not required if you have configured an explicit schema. Use this parameter if you need to override the configured schema qualifier. If no -dbSchema is provided, the schema name from the data source is used, and if no explicit schema is configured for the data source, the implicit (default) schema is used:

- For DB2 the implicit schema is the user ID used to connect to the database, so if you specify the -dbUser parameter, you might need to specify the -dbSchema parameter to prevent the wrong user ID being used as the implicit schema.
- For Microsoft SQL Server, the implicit schema is "dbo".
- For Derby, the implicit schema is "APP".

-slice *slice*
This optional parameter specifies the transaction size, which can be between `10` and `50000`. The default value is 500. Optimum values depend on many factors including the size of the database objects, and the size of the transaction log. In general, smaller values tend to result in longer migration times.

-force TRUE

Attention: You should only use this optional parameter if you need to run the script again, and you are sure that no other instance of the migration script is still running. If more than one instance runs at the same time, the database might become inconsistent, and require a restore from your backup
You might need to use this option if the migration script stopped before it completed the migration, for example, because the connection was lost to the database server. Using this option is the only way to run the script again, the script will attempt to continue migrating the data from the point that it had reached. Using this option is the only way to prevent the com.ibm.bpe.api.DatabaseDataMigrationAlreadyRunningException exception:

```
CWWBB0654E: The data migration has already been started.
```

Example

To invoke the script to convert the runtime data in the database used by the Business Process Choreographer configuration on cluster “hamlet”, using a transaction slice size of 5,000, the database user “Sam”, password “secret”, and sending the trace information to a file named “migrateDB.traceout” in a typical location, enter one of the following commands.

On Linux and UNIX platforms:

```
wsadmin.sh -conntype NONE
        -tracefile install_root/profiles/ProcSrv01/logs/migrateDB.traceout
        -f install_root/ProcessChoreographer/admin/migrateDB.py
        -cluster hamlet -dbUser Sam -dbPassword secret -slice 5000
```

On i5/OS platforms:

```
wsadmin -conntype NONE
        -tracefile install_root/profiles/ProcSrv01/logs/migrateDB.traceout
        -f install_root/ProcessChoreographer/admin/migrateDB.py
        -cluster hamlet -dbUser Sam -dbPassword secret -slice 5000
```

On Windows platforms:

```
wsadmin -conntype NONE
        -tracefile install_root\profiles\ProcSrv01\logs\migrateDB.traceout
        -f install_root\ProcessChoreographer\admin\migrateDB.py
        -cluster hamlet -dbUser Sam -dbPassword secret -slice 5000
```

Recovering after problems

If the script fails to complete the migration with the com.ibm.bpe.api.DatabaseDataMigrationAlreadyRunningException exception, you can restart the script using the `-force` option. Before you restart the script again, ensure that no other instance of the migration script is still running.

Upgrading the Business Space database manually

After migrating the server from version 6.1.2, you must manually upgrade the Business Space database to a new database schema before you start the version 6.2 server.

Before you begin

- You should have already run the migration wizard or migration scripts to migrate the server, or in the case of a cluster, the servers in the cluster.
- Make sure the server or, if applicable, servers in the cluster remain stopped (do not start them after the migration wizard or scripts have run before completing the database upgrade).

About this task

After migrating WebSphere Process Server from version 6.1.2 to version 6.2, the Business Space database is still configured to be used with Business Space 6.1.2. You must upgrade the Business Space database before it is accessed by Business Space version 6.2. This step consists of running two database scripts. The first script upgrades the database schema. The second script modifies data in the Business Space database to prepare it for use by Business Space version 6.2.

Note: Before running the scripts, you need to edit them and replace all occurrences of the `@SCHEMA@` string with the schema name of the Business Space database. The default schema name is `IBMBUSSP`. When running the script, the database user must have sufficient authorization to modify the database.

Procedure

1. Make sure that you are using a user ID with sufficient authority to update the database schema.
2. Back up the database.
3. Locate the directory where the database scripts are located:
   - Windows On Windows platforms: *install_root*\dbscripts\BusinessSpace\ *database_type*
   - Linux UNIX On Linux and UNIX platforms: *install_root*/dbscripts/ BusinessSpace /*database_type*

   Where:

   install_root

   is the root directory into which WebSphere Process Server version 6.2 has been installed.

   database_type

   is a name corresponding to the database product you are using. Applicable database types and their directory names are as follows:

Database type	**Directory name**
DB2 on iSeries	DB2iSeries
DB2 Universal Database (for all operating systems except z/OS and i5/OS)	DB2
DB2 for z/OS Version 8.x	DB2zOSV8 Use scripts in this directory if your initial database configuration used DB2 z/OS v8 or later (uses long table names) **Note:** If you used DB2 z/OS v7 during your initial database configuration, refer to the relevant technote at the WebSphere Process Server support site.
DB2 for z/OS Version 9.x	DB2zOSV9 Use the scripts in this directory if your initial database configuration used DB2 z/OS v9 and a DB2 z/OS v9 key **Note:** If you used DB2 z/OS v7 during your initial database configuration, refer to the relevant technote at the WebSphere Process Server support site.

Database type	**Directory name**
Derby	Derby **Note:** If you have an existing Cloudscape database for a previous (6.0.x) installation of WebSphere Process Server, the database must first be upgraded from a Cloudscape database to a Derby database. See “Migrating Cloudscape databases” on page 128. Then, once you have a Derby database that corresponds to that same (6.0.x) WebSphere Process Server version, you must then upgrade the database to correspond to WebSphere Process Server version 6.2 with the scripts in this directory.

Oracle	Oracle
	Note: If you have not applied IFix001 for Business Space version 6.1.2, uncomment out the following lines from upgradeSchema612_BusinessSpace.sql before running the script:
	`-- ALTER TABLE @SCHEMA@.PAGE`
	`--       RENAME COLUMN RESTRICTEDD TO RESTRICTED;`

For specific database product names and versions see Database configurations.

4. Locate the migration scripts for the database and current schema version, where *schema_version* has the value 612 for version 6.1.2. For example, to upgrade the Business Space database schema for DB2 from WebSphere Process Server version 6.1.2 to version 6.2, you need *install_root*/dbscripts/BusinessSpace/DB2/upgradeSchema612_BusinessSpace.sql.

   Note: You also need the upgradeData612_BusinessSpace.sql script.
5. Copy the appropriate scripts from the directory in which you found them to the system on which the database is running.
6. Modify the SQL scripts upgradeSchema612_BusinessSpace.sql and upgradeData612_BusinessSpace.sql to meet your requirements. For example, you must change the schema name of the database, and you might need to change a user name, password, or file path.
7. Using your database client, connect to the database. This is to ensure that it can connect.
8. Run your version of the upgrade SQL scripts. For information on how to execute a .sql script with your database, refer to the documentation for your database product.
   a. Run upgradeSchema612_BusinessSpace.sql to upgrade the database schema.
   b. Run upgradeData612_BusinessSpace.sql to modify the data in the Business Space database.
9. If there are any errors, or failure is indicated in your database client output, fix the reported errors and retry step 8.

Results

The database has been updated, and is ready for use by Business Space version 6.2. After you run the script to upgrade the database schema, the database can no longer be used by Business Space version 6.1.2.

Related tasks

Postmigration tasks for Business Space powered by WebSphere
After migrating WebSphere Process Server from version 6.1.2 to version 6.2, you must perform some additional tasks before you start your servers or clusters.

Migrating stand-alone servers

Migrate a stand-alone WebSphere Process Server server, choosing from several methods depending on your needs.

Before you begin

See "Overview of migrating" on page 1 and "Premigration considerations for WebSphere Process Server" on page 4.

Select the appropriate subtopic for information on how to migrate a WebSphere Process Server stand-alone server from an older version to a newer version of WebSphere Process Server.

Tip: For help in troubleshooting problems when migrating, see "Troubleshooting version-to-version migration" on page 140.

Migrating a stand-alone server using the migration wizard

Migrate a stand-alone server from an older version to a newer version of WebSphere Process Server using the migration wizard.

Before you begin

Note: The migration wizard cannot run in a non-graphical environment. Examples of non-graphical environments include the **i5/OS platform** or telnet sessions. If you want to run migration in a non-graphical environment, use the WBIPreUpgrade and WBIPostUpgrade commands.

Note: The migration wizard supports only WebSphere Process Server profiles. If you have WebSphere Application Server profiles, you must use the migration commands.

Make sure that the following conditions are met before you start the migration process:

- Ensure that your system meets all hardware and software requirements for installing WebSphere Process Server version 6.2, and that you have enough space (including temporary space) for your installation. See http://www.ibm.com/support/docview.wss?uid=swg27006205 for more information.
- If you are migrating on the same physical computer system on which the older version of WebSphere Process Server resides, you have installed the new version of WebSphere Process Server side-by-side the old version on the same system.

  Note: If you plan to migrate to a new physical computer system then use the alternate procedure described in "Migrating to a remote system" on page 58. Or, if you plan to upgrade the version of the operating system on the computer that is running WebSphere Process Server, then use the alternate procedure described in "Migrating from an operating system that is no longer supported" on page 64.

- The profile in the older WebSphere Process Server which you are migrating is stand-alone (not federated to a cell).
- Sufficient disk space is available for the migrated profile and its backup. See “Premigration considerations for WebSphere Process Server” on page 4 for disk space requirements.

Make sure that you have completed the following tasks before you start the migration process:

- Back up the databases that support version 6.0.2.x or version 6.1.x WebSphere Process Server components.

Procedure

1. Log on as the root user on a Linux or UNIX system, or as a member of the Administrator group on a Windows system.
2. Stop the version 6.1.x or version 6.0.2.x server if it is running on the node to be migrated. Use the stopServer command from the *profile_dir*/bin directory for the profile of the affected server, or stop the server from the profile’s First steps console.

   For more information about the stopServer command see the stopServer command. Use the following syntax:

   - Linux UNIX **On Linux and UNIX platforms:** `profile_root/bin/stopServer.sh server_name`
   - Windows **On Windows platforms:** `profile_root\bin\stopServer.bat server_name`

   If security is enabled, use one of the following commands instead. The user name provided must be a member of the operator or administrator role.

   - Linux UNIX **On Linux and UNIX platforms:** `profile_root/bin/stopServer.sh server_name -username user_ID -password password`
   - Windows **On Windows platforms:** `profile_root\bin\stopServer.bat server_name -username user_ID -password password`

   On the Windows operating system, even if security is enabled, the -username and -password parameters do not have to be specified if the server is running as a Windows service. In this case, the parameters are automatically passed into the script that the Windows service uses to shut down the system.

   Note: Before you start the migration process, you must stop the server from which you are migrating. It is not necessary to have that server running in order to migrate its configuration. The migration tools can retrieve all of the configuration data while the server is stopped.
3. Optional: Create a new version 6.2 profile as the migration target using the **Create new profile** option in the migration wizard. For more information about creating target profiles, refer to “Target profile considerations” on page 9.
4. Identify, in advance, the pre-existing information required for the migration, as listed below:

   Installation root directory

   See WBIPreUpgrade command-line utility for a description of the currentWebSphereDirectory parameter.

   Migration backup directory name

   See WBIPreUpgrade command-line utility for a description of the backupDirectory parameter.

Administrative security user name (required if adminstrative security is configured)
See WBIPostUpgrade command-line utility for a description of the -username parameter.

Administrative security password (required if adminstrative security is configured)
See WBIPostUpgrade command-line utility for a description of the -password parameter.

Source profile name
See WBIPostUpgrade command-line utility for a description of the -oldProfile parameter.

Target profile name
See WBIPostUpgrade command-line utility for a description of the -profileName parameter.

Port value assignments (optional)
See WBIPostUpgrade command-line utility for a description of the -replacePorts and -portBlock parameters.

Note: This applies only if you are migrating from version 6.0.2.x toversion 6.2.

5. Invoke the migration wizard.

   Invoke the migration wizard in one of the following ways:

   - From the WebSphere Process Server First Steps console, select **Migration wizard**.
   - Run one of the following scripts (depending upon your operating system) stored in the *install_dir*/bin directory:
     - Linux UNIX **On Linux and UNIX platforms:** `wbi_migration.sh`
     - Windows **On Windows platforms:** `wbi_migration.bat`

     Note: You can optionally change the default trace setting (`*=all=enabled:com.ibm.ws.migration.common.*=all=disabled`) when invoking the migration wizard. The default trace setting enables tracing on only certain classes, but you can change the default to either enable full tracing or disable all tracing.

     - To enable full tracing, run one of the following scripts to invoke the migration wizard, depending on your operating system:
       - Linux UNIX **On Linux and UNIX platforms:** `wbi_migration.sh -W -migrationPanel.traceString="*=all=enabled"`
       - Windows **On Windows platforms:** `wbi_migration.bat -W -migrationPanel.traceString="*=all=enabled"`
     - To disable all tracing, run one of the following scripts to invoke the migration wizard, depending on your operating system:
       - Linux UNIX **On Linux and UNIX platforms:** `wbi_migration.sh -W -migrationPanel.traceString="*=all=disabled"`
       - Windows **On Windows platforms:** `wbi_migration.bat -W -migrationPanel.traceString="*=all=disabled"`

   For information about what processing the migration wizard actually performs, see “What the migration wizard does” on page 19.

6. Follow the prompts for the migration wizard as described in “Running the migration wizard” on page 21.

7. If required, manually update the databases used by WebSphere Process Server. Some database changes required by new versions of WebSphere Process Server are made automatically. However, if you have Business Process Choreographer or Business Space configured, or if the server has insufficient permissions to access the database schema, or other database-specific requirements are not met, you must update the database manually. For more information see “Upgrading databases for migration” on page 34.

Results

You have now migrated your stand-alone server.

What to do next

Verify that the migration has been successful. If your server has Business Process Choreographer configured, see “Postmigration tasks for Business Process Choreographer” on page 123. If your server has Business Space configured, see “Postmigration tasks for Business Space powered by WebSphere” on page 127. Finally, perform the checks described in “Postmigration tasks for WebSphere Process Server” on page 122.

Related concepts

Premigration considerations for Business Process Choreographer
If your servers run Business Process Choreographer, you should be aware of certain things that you need to plan and take into consideration before you migrate Business Process Choreographer.

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

Migration wizard
The version-to-version migration wizard is a graphical interface that guides you through migrating from an older version to a newer version of WebSphere Process Server.

Related tasks

Running the migration wizard
Run the migration wizard on AIX, HP-UX, Linux, Solaris, or Windows systems to migrate WebSphere Process Server.

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

Postmigration tasks for WebSphere Process Server
After migration, you should check some configuration settings. You might need to change them, or further configure the version 6.2 server.

Upgrading databases for migration
In conjunction with migration, the database schema of some WebSphere Process Server components must be upgraded. This can occur automatically but in some cases you must upgrade the schema manually.

Creating profiles
Learn how to create new WebSphere Enterprise Service Bus or WebSphere Process Server profiles. You can create profiles from a command line by using the manageprofiles command, or interactively by using the Profile Management Tool graphical user interface (GUI).

Migrating to a remote system
Use the migration tools to migrate from an older version on one system to a newer version of WebSphere Process Server on a different, remote system. (Stand-alone servers only.)

Migrating from an operating system that is no longer supported
Use the migration tools to migrate an earlier WebSphere Process Server release that is running on an operating system that the newer version does not support. (Stand-alone servers only.)

Related reference

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

WBIPostUpgrade command-line utility
Use the WBIPostUpgrade command for WebSphere Process Server to retrieve the profile configuration that was saved by the WBIPreUpgrade command at the *backupDirectory* that you had specified.

WBIPreUpgrade command-line utility
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 stand-alone server using command-line tools

Migrate a stand-alone server from an older version to a newer version of WebSphere Process Server using the command-line tools.

Before you begin

Note: When you migrate using the command-line tools, you can migrate either a WebSphere Process Server profile or a WebSphere Application Server profile.

Make sure that the following conditions are met before you start the migration process:

- Ensure that your system meets all hardware and software requirements for installing WebSphere Process Server version 6.2, and that you have enough space (including temporary space) for your installation. See http://www.ibm.com/support/docview.wss?uid=swg27006205 for more information.
- If you are migrating on the same physical computer system on which the older version of WebSphere Process Server resides, you have installed the new version of WebSphere Process Server side-by-side the old version on the same system.

  Note: If you plan to migrate to a new physical computer system then use the alternate procedure described in "Migrating to a remote system" on page 58. Or, if you plan to upgrade the version of the operating system on the computer that is running WebSphere Process Server, then use the alternate procedure described in "Migrating from an operating system that is no longer supported" on page 64.
- The profile in the older WebSphere Process Server which you are migrating is stand-alone (not federated to a cell).
- Sufficient disk space is available for the migrated profile and its backup. See "Premigration considerations for WebSphere Process Server" on page 4 for disk space requirements.

Make sure that you have completed the following tasks before you start the migration process:

- Back up the databases that support version 6.0.2.x or version 6.1.x WebSphere Process Server components.

Procedure

1. Log on using one of the following procedures, depending on your operating system.
   - i5/OS **On i5/OS platforms:** Log on with an i5/OS user profile that has `*SECOFR` user class or `*ALLOBJ` special authority.
   - Linux UNIX **On Linux and UNIX platforms:** Log on as the root user.
   - Windows **On Windows platforms:** Log on as a member of the Administrator group.

2. Stop the version 6.1.x or 6.0.2.x server if it is running on the node to be migrated. Use the stopServer command from the *profile_dir*/bin directory for the profile of the affected server, or stop the server from the profile's First steps console.

   For more information about the stopServer command see the stopServer command. Use the following syntax:

   Note: On i5/OS platforms, you must run the scripts under QSHELL. To start a QSHELL session, open a CL command prompt and type QSH.

   - i5/OS **On i5/OS platforms:** *profile_root*/bin/stopServer *server_name*
   - Linux UNIX **On Linux and UNIX platforms:** *profile_root*/bin/stopServer.sh *server_name*
   - Windows **On Windows platforms:** *profile_root*\bin\stopServer.bat *server_name*

   If security is enabled, use one of the following commands instead. The user name provided must be a member of the operator or administrator role.

   - i5/OS **On i5/OS platforms:** *profile_root*/bin/stopServer *server_name* -username *user_ID* -password *password*
   - Linux UNIX **On Linux and UNIX platforms:** *profile_root*/bin/stopServer.sh *server_name* -username *user_ID* -password *password*
   - Windows **On Windows platforms:** *profile_root*\bin\stopServer.bat *server_name* -username *user_ID* -password *password*

   On the Windows operating system, even if security is enabled, the -username and -password parameters do not have to be specified if the server is running as a Windows service. In this case, the parameters are automatically passed into the script that the Windows service uses to shut down the system.

   Note: Stop the server before beginning the migrating process. By default, all servers on the node are stopped before the migration completes.
3. Run the WBIPreUpgrade command, specifying the migration backup directory name and the existing WebSphere Process Server directory name. The WBIPreUpgrade tool saves selected files from the *install_root* and *profile_root* directories to a backup directory you specify. See the WBIPreUpgrade command-line utility for details.
4. Run the WBIPostUpgrade command, specifying the migration backup directory. The WBIPostUpgrade tool restores the environment in the backup directory into the new WebSphere Process Server stand-alone server installation. See the WBIPostUpgrade command-line utility for details.

   Important: Use the `-createTargetProfile` parameter when invoking WBIPostUpgrade. This option creates a matching required new target profile for migration. For more information about target profiles, refer to "Target profile considerations" on page 9.

   Note: i5/OS If you are migrating on an **i5/OS platform**, the target profile name must match the profile name of the source profile being migrated.
5. If required, manually update the databases used by WebSphere Process Server. Some database changes required by new versions of WebSphere Process Server are made automatically. However, if you have Business Process Choreographer or Business Space configured, or if the server has insufficient permissions to access the database schema, or other database-specific requirements are not

met, you must update the database manually. For more information see "Upgrading databases for migration" on page 34.

Results

You have now migrated your stand-alone server.

What to do next

Verify that the migration has been successful. If your server has Business Process Choreographer configured, see "Postmigration tasks for Business Process Choreographer" on page 123. If your server has Business Space configured, see "Postmigration tasks for Business Space powered by WebSphere" on page 127. Finally, perform the checks described in "Postmigration tasks for WebSphere Process Server" on page 122.

Related concepts

Premigration considerations for Business Process Choreographer
If your servers run Business Process Choreographer, you should be aware of certain things that you need to plan and take into consideration before you migrate Business Process Choreographer.

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

Migration wizard
The version-to-version migration wizard is a graphical interface that guides you through migrating from an older version to a newer version of WebSphere Process Server.

Related tasks

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

Postmigration tasks for WebSphere Process Server
After migration, you should check some configuration settings. You might need to change them, or further configure the version 6.2 server.

Upgrading databases for migration
In conjunction with migration, the database schema of some WebSphere Process Server components must be upgraded. This can occur automatically but in some cases you must upgrade the schema manually.

Creating profiles
Learn how to create new WebSphere Enterprise Service Bus or WebSphere Process Server profiles. You can create profiles from a command line by using the manageprofiles command, or interactively by using the Profile Management Tool graphical user interface (GUI).

Related reference

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

WBIPostUpgrade command-line utility
Use the WBIPostUpgrade command for WebSphere Process Server to retrieve the profile configuration that was saved by the WBIPreUpgrade command at the *backupDirectory* that you had specified.

WBIPreUpgrade command-line utility
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 to a remote system

Use the migration tools to migrate from an older version on one system to a newer version of WebSphere Process Server on a different, remote system. (Stand-alone servers only.)

Before you begin

Note: This procedure is supported for stand-alone servers only.

Make sure that the following conditions are met before you start the migration process:

- Your target system meets all the hardware and software requirements for the new version of WebSphere Process Server.
- Sufficient disk space is available for the migrated profile and its backup. See "Premigration considerations for WebSphere Process Server" on page 4 for disk space requirements.

Make sure that you have completed the following tasks before you start the migration process:

- Back up the databases that support version 6.0.2.x or version 6.1.x WebSphere Process Server components.

See "Overview of migrating" on page 1 and "Premigration considerations for WebSphere Process Server" on page 4.

Typically, you can use the migration tools to upgrade from an older version to a newer version on the same system. However, some scenarios require that you migrate the old version's configuration on one system to the new version of WebSphere Process Server on a different system. One of these scenarios is when you install new systems for your environment based on the newer version but need to migrate your existing older configuration from other systems.

For help in troubleshooting problems when migrating, see "Troubleshooting version-to-version migration" on page 140.

About this task

The **WBIPreUpgrade** command saves the existing older version configuration into a migration-specific backup directory. The **WBIPostUpgrade** command uses this directory to add the old configuration settings to the newer version environment.

Procedure

1. Copy the WBIPreUpgrade utility and its related files from WebSphere Process Server version 6.2 onto the source system. To do this, use one of the following procedures.
   a. On the target system, create a .zip or .tar.gz file of the installation files to copy to the source system. To do this, use the following procedure.

      Note: You must already have WebSphere Process Server version 6.2 installed on the target system to use this procedure.

      1) Navigate to the WPS62_HOME/util/migration directory on the target system.
      2) Run one of the following commands, depending on your operation system.
         - Linux UNIX **On Linux and UNIX platforms**: `createRemoteMigrationImage.sh` *`full_path_name_of_the_new_.tar.gz_file`*

           Example: `createRemoteMigrationImage.sh /tmp/migrationImage.tar.gz`

        - Windows **On Windows platforms**: `createRemoteMigrationImage.bat` *`full_path_name_of_the_new_.zip_file`*

          Example: `createRemoteMigrationImage.bat C:\migrationImage.zip`

    3) Copy the migration image .zip file from the target system to the source system (the system that has version 6.1.x or 6.0.2.x of WebSphere Process Server installed on it).
    4) Unzip the migration image file to a new directory called migration_copy.

       Note: This directory can be any name you choose. We are using ″migration_copy″ here for explanatory purposes.

  b. Copy the files from the JDK and migration directories of the WebSphere Process Server version 6.2 DVD. To do this, use the following procedure. Copy the migration and JDK directories from the WPS/ESB Version 6.2 DVD to a new directory . For instance, we use (migration_copy for explanatory purposes.)
    1) Create a new directory on the source system called migration_copy.

       Note: This directory can be any name you choose. We are using ″migration_copy″ here for explanatory purposes.
    2) Copy the migration and JDK directories in the new directory.

  You should now have the following directory structure on your source system:

```
migration_copy/
   migration/
   JDK/
```

2. Navigate to the migration_copy/migration/bin directory.
3. On the source system, save the current configuration using the **WBIPreUpgrade** script from the migration_copy/bin directory. Save the configuration in the migration-specific backup directory on the system hosting the older version (source system). See the WBIPreUpgrade command-line utility for details. Use one of the following scripts, depending on your operating system.
   - Linux UNIX **On Linux/UNIX platforms:** ./WBIPreUpgrade.sh */filepath/migration_specific_backup currentWebSphereDirectory*
   - Windows **On Windows platforms:** WBIPreUpgrade C:\filepath\\ *migration_specific_backup currentWebSphereDirectory*

   The **WBIPreUpgrade** command provides status to the screen and to log files in the *migration_specific_backup*/logs directory. Log file names start with the text `WBIPreUpgrade` and include a date and timestamp.
4. Copy the *migration_specific_backup* directory from the source system to the target system.

   Use the **ftp** command, shared storage, or some other mechanism to copy the directory to the new system.
5. On the target system, add the configuration from the older version of WebSphere Process Server configuration to the newer version's configuration using the **WBIPostUpgrade** command. Use the **WBIPostUpgrade** command in the *install_root*/bin directory of the new installation to add the older version's configuration (which you copied into a directory on the new system in step 4) to the newer version's configuration.

   See the WBIPostUpgrade command-line utility for details. Use one of the following scripts, depending on your operating system.

Important: Use the `-createTargetProfile` parameter when invoking WBIPostUpgrade. This option creates a matching required new target profile for migration. For more information about target profiles, refer to “Target profile considerations” on page 9.

Note: In the following scripts, *-profileName* refers to the target profile.

- Linux UNIX **On Linux/UNIX platforms:** ./WBIPostUpgrade.sh /*filepath*/*migration_specific_backup*/-profileName*profileName*
- Windows **On Windows platforms:** WBIPostUpgrade C:\filepath\migration_specific_backup/-profileName*profileName*

The **WBIPostUpgrade** command provides status to the screen and to log files in the *migration_specific_backup*/logs directory. Log file names start with the text `WBIPreUpgrade` and include a date and timestamp.

6. If required, manually update the databases used by WebSphere Process Server. Some database changes required by new versions of WebSphere Process Server are made automatically. However, if you have Business Process Choreographer or Business Space configured, or if the server has insufficient permissions to access the database schema, or other database-specific requirements are not met, you must update the database manually. For more information see “Upgrading databases for migration” on page 34.
7. If the databases supporting WebSphere Process Server resided on the local system in the original setup, they will now be remote to WebSphere Process Server after moving to a different system. Therefore, if you used a type 2 JDBC driver you may need to switch to a type 4 driver now.
8. Modify the configuration using the newer WebSphere Process Server version’s administrative console.
   a. Change user IDs and passwords to match security requirements.

      You might have to change user IDs and passwords if they are not identical to those in use on the system hosting the older version.
   b. Change other system-specific information.

      The configuration might refer to other software products or configurations that do not exist on the new system. For example, the old system might have a database. Modify the data source to point to the database on the old system.

Results

You have migrated WebSphere Process Server from the older version to a remote system hosting the newer version.

What to do next

Verify that the migration has been successful. If your server has Business Process Choreographer configured, see “Postmigration tasks for Business Process Choreographer” on page 123. If your server has Business Space configured, see “Postmigration tasks for Business Space powered by WebSphere” on page 127. Finally, perform the checks described in “Postmigration tasks for WebSphere Process Server” on page 122.

Related concepts

Premigration considerations for Business Process Choreographer
If your servers run Business Process Choreographer, you should be aware of certain things that you need to plan and take into consideration before you migrate Business Process Choreographer.

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

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

Related tasks

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

Postmigration tasks for WebSphere Process Server
After migration, you should check some configuration settings. You might need to change them, or further configure the version 6.2 server.

Creating profiles
Learn how to create new WebSphere Enterprise Service Bus or WebSphere Process Server profiles. You can create profiles from a command line by using the manageprofiles command, or interactively by using the Profile Management Tool graphical user interface (GUI).

Migrating from an operating system that is no longer supported
Use the migration tools to migrate an earlier WebSphere Process Server release that is running on an operating system that the newer version does not support. (Stand-alone servers only.)

Related reference

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

WBIPreUpgrade command-line utility
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-line utility
Use the WBIPostUpgrade command for WebSphere Process Server to retrieve the profile configuration that was saved by the WBIPreUpgrade command at the *backupDirectory* that you had specified.

Related information

Installing and configuring WebSphere Process Server
This section describes how to prepare for, install, and configure an installation of IBM WebSphere Process Server. Instructions are provided for Linux, i5/OS, UNIX, and Windows systems.

Migrating from a 32-bit Windows platform to a 64–bit Windows platform

If you have a WebSphere Process Server version 6.1.x or 6.0.2.x installation that is installed on a Windows 32-bit operating system, you can migrate your existing stand-alone profiles to a 64-bit version of WebSphere Process Server version 6.2 installed on a Windows 64-bit operating system. before performing the migration to WebSphere Process Server version 6.2.

Before you begin

Note: This procedure is supported for stand-alone servers only.

Make sure that the following conditions are met before you start the migration process:

- Your target system meets all the hardware and software requirements for the new version of WebSphere Process Server.
- Sufficient disk space is available for the migrated profile and its backup. See "Premigration considerations for WebSphere Process Server" on page 4 for disk space requirements.

Make sure that you have completed the following tasks before you start the migration process:

- Back up the databases that support version 6.0.2.x or version 6.1.x WebSphere Process Server components.

See "Overview of migrating" on page 1 and "Premigration considerations for WebSphere Process Server" on page 4.

Typically, you can use the migration tools to upgrade from an older version to a newer version on the same system. However, some scenarios require that you migrate the old version's configuration on one system to the new version of WebSphere Process Server on a different system. One of these scenarios is when you install new systems for your environment based on the newer version but need to migrate your existing older configuration from other systems.

For help in troubleshooting problems when migrating, see "Troubleshooting version-to-version migration" on page 140.

About this task

The **WBIPreUpgrade** command saves the existing older version configuration into a migration-specific backup directory. The **WBIPostUpgrade** command uses this directory to add the old configuration settings to the newer version environment.

Procedure

1. Copy the WBIPreUpgrade utility and its related files from the 32–bit version of WebSphere Process Server version 6.2 onto the source (32–bit) system. To do this, use one of the following procedures.
   a. On the target system, create a .zip or .tar.gz file of the installation files to copy to the source system. To do this, use the following procedure.

      Note: You must already have the 32–bit version of WebSphere Process Server version 6.2 installed on the target system to use this procedure.

      1) Navigate to the WPS62_HOME/util/migration directory on the target system.
      2) Run one of the following commands, depending on your operation system.

        - Linux UNIX **On Linux and UNIX platforms**: `createRemoteMigrationImage.sh` *`full_path_name_of_the_new_.tar.gz_file`*

          Example: `createRemoteMigrationImage.sh /tmp/migrationImage.tar.gz`
        - Windows **On Windows platforms**: `createRemoteMigrationImage.bat` *`full_path_name_of_the_new_.zip_file`*

          Example: `createRemoteMigrationImage.bat C:\migrationImage.zip`
     3) Copy the migration image .zip file from the target system to the source system (the system that has version 6.1.x or 6.0.2.x of WebSphere Process Server installed on it).
     4) Unzip the migration image file to a new directory called migration_copy.

        Note: This directory can be any name you choose. We are using ″migration_copy″ here for explanatory purposes.
   b. Copy the files from the JDK and migration directories of the 32–bit WebSphere Process Server version 6.2 DVD. To do this, use the following procedure. Copy the migration and JDK directories from the WPS/ESB Version 6.2 DVD to a new directory . For instance, we use (migration_copy for explanatory purposes.)
     1) Create a new directory on the source system called migration_copy.

        Note: This directory can be any name you choose. We are using ″migration_copy″ here for explanatory purposes.
     2) Copy the migration and JDK directories in the new directory.

   You should now have the following directory structure on your source system:

```
migration_copy/
   migration/
   JDK/
```

2. Copy the backup directory of the WebSphere Process Server version 6.1.x or 6.0.2.x installation to another system.
3. Install Windows 2003 64-bit operating system on the system where you want to perform the migration.
4. Install WebSphere Process Server version 6.2 on the system with the new Windows 2003 64-bit operating system.
5. Run the WBIPostUpgrade command against the backup directory to complete the migration.

Results

You migrated to WebSphere Process Server version 6.2 from a version 6.1.x or 6.0.2.x installation that was installed on a Windows 32-bit operating system.

Migrating from an operating system that is no longer supported

Use the migration tools to migrate an earlier WebSphere Process Server release that is running on an operating system that the newer version does not support. (Stand-alone servers only.)

Before you begin

Note: This procedure is supported for stand-alone servers only.

Note: When you migrate using the command-line tools, you can migrate either a WebSphere Process Server profile or a WebSphere Application Server profile.

Make sure that the following conditions are met before you start the migration process:

- Your system meets all the hardware and software requirements for the new version of WebSphere Process Server.
- Sufficient disk space is available for the migrated profile and its backup. See “Premigration considerations for WebSphere Process Server” on page 4 for disk space requirements.

Make sure that you have completed the following tasks before you start the migration process:

- Back up the databases that support version 6.0.2.x or version 6.1.x WebSphere Process Server components.

See “Overview of migrating” on page 1 and “Premigration considerations for WebSphere Process Server” on page 4.

For help in troubleshooting problems when migrating, see “Troubleshooting version-to-version migration” on page 140.

Procedure

1. Copy the WBIPreUpgrade utility and its related files from WebSphere Process Server version 6.2 onto the source system. To do this, use one of the following procedures.
   a. On the target system, create a .zip or .tar.gz file of the installation files to copy to the source system. To do this, use the following procedure.

      Note: You must already have WebSphere Process Server version 6.2 installed on the target system to use this procedure.

      1) Navigate to the WPS62_HOME/util/migration directory on the target system.
      2) Run one of the following commands, depending on your operation system.
         - Linux UNIX **On Linux and UNIX platforms**: `createRemoteMigrationImage.sh` *`full_path_name_of_the_new_.tar.gz_file`*

           Example: `createRemoteMigrationImage.sh /tmp/migrationImage.tar.gz`
         - Windows **On Windows platforms**: `createRemoteMigrationImage.bat` *`full_path_name_of_the_new_.zip_file`*

           Example: `createRemoteMigrationImage.bat C:\migrationImage.zip`
      3) Copy the migration image .zip file from the target system to the source system (the system that has version 6.1.x or 6.0.2.x of WebSphere Process Server installed on it).
      4) Unzip the migration image file to a new directory called migration_copy.

    Note: This directory can be any name you choose. We are using ″migration_copy″ here for explanatory purposes.

   b. Copy the files from the JDK and migration directories of the WebSphere Process Server version 6.2 DVD. To do this, use the following procedure. Copy the migration and JDK directories from the WPS/ESB Version 6.2 DVD to a new directory . For instance, we use (migration_copy for explanatory purposes.)

      1) Create a new directory on the source system called migration_copy.

         Note: This directory can be any name you choose. We are using ″migration_copy″ here for explanatory purposes.

      2) Copy the migration and JDK directories in the new directory.

   You should now have the following directory structure on your source system:

```
migration_copy/
   migration/
   JDK/
```

2. Navigate to the migration_copy/migration/bin directory.
3. Save the current configuration using the **WBIPreUpgrade** script from the migration_copy/bin directory. Save the configuration in the migration-specific backup directory on the system hosting the older version. See the WBIPreUpgrade command-line utility for details. Use one of the following scripts, depending on your operating system.

   - i5/OS **On i5/OS platforms:** ./WBIPreUpgrade /filepath/ *migration_specific_backup currentWebSphereDirectory*

     Note: On i5/OS platforms, the user profile must have *ALLOBJ authority. Also, you must run i5/OS the scripts under QSHELL. To start a QSHELL session, open the CL command prompt and type `QSH`.

   - Linux UNIX **On Linux/UNIX platforms:** ./WBIPreUpgrade.sh */filepath*/*migration_specific_backup currentWebSphereDirectory*
   - Windows **On Windows platforms:** WBIPreUpgrade C:\filepath\\ *migration_specific_backup currentWebSphereDirectory*

   The **WBIPreUpgrade** command provides status to the screen and to log files in the *migration_specific_backup*/logs directory. Log file names start with the text `WBIPreUpgrade` and include a date and timestamp.
4. Shut down the older WebSphere Process Server release by stopping all server nodes in the configuration.
5. Use the following procedure to back up your system, depending on your operating system:

   - i5/OS **On i5/OS platforms**: Perform the following backup tasks to migrate from i5/OS V5R3 to i5/OS V5R4:
     - Back up each of the WebSphere Process Server or WebSphere Application Server stand-alone profiles that were configured on i5/OS V5R3 by using the WBIPreUpgrade command
     - Back up any database schemas associated with each of the profiles by using i5/OS native `save` commands for database collections. For example: `crtsavf` and `savlib`.
     - Back up anything special that you might have customized in these profiles.

    - Keep track of any special authorities you might have assigned to i5/OS user profiles to the configuration directories/files under each version 6.1.x profile. You can manually restore these private authorities under the target profiles after migrating to WebSphere Process Server version 6.2.
  - Linux UNIX Windows **On Linux, UNIX, and Windows platforms**: Compress the backup directory (with a utility such as TAR or ZIP), and use FTP to copy the file to another system.

6. Use the following procedure to install the new operating system, depending on which operating system you are using:
   - i5/OS **On i5/OS platforms**: Install the new i5/OS V5R4 or i5/OS V6R1 operating system, using the following procedure.
     a. Slip-install either i5/OS V5R4 or i5/OS V6R1 on top of i5/OS V5R3, using the instructions in the Memorandum to Users for the operating system you are installing:
        - If you are installing i5/OS V5R4, use the following instructions: http://www-912.ibm.com/s_dir/sline003.nsf/3a8f58452f9800bc862562900059e09e/bdb2077acff30ff28625710f005ca12f?OpenDocument.
        - If you are installing i5/OS V6R1, use the following instructions: http://www-912.ibm.com/s_dir/sline003.nsf/PSP Number View/SF98026 . Also see the ″What's New″ section under July 08, 2008 in the following link: http://www-03.ibm.com/systems/i/software/websphere/news/sitenews.html. This section describes the changes for WebSphere Application Server version 6.1 in V6R1 especially regarding HTTP Server plug-in changes.

     b. Apply the latest PTF (Program temporary fix) for the operating system that you slip-installed.

     c. Apply the latest PTF groups for supporting products, such as: DB2 for i5/OS, Java, IBM Tookbox for Java, and IBM HTTP Server.
   - Linux UNIX Windows **On Linux. UNIX, and Windows platforms**: Install the new operating system, keeping the same host name. If possible, keep the system name and passwords the same as on the old system. Place any database files related to applications that you are migrating in the same path as on the previous system. In general, try to keep paths the same. If you must change paths or names, make any changes with the administrative console before running the **WBIPostUpgrade** command as described in a later step.

7. Use FTP to copy the backup directory from the other system, and extract it.

8. Install the newer version of WebSphere Process Server.

   See Installing and configuring WebSphere Process Server.

9. Run the **WBIPostUpgrade** command from the newer version*install_root*/bin directory.

   Specify the copy of the backup directory that you made in step 7. See WBIPostUpgrade command-line utility for the correct command syntax. For example:

   ```
   install_root\bin\WBIPostUpgrade wbi_installation\migration
   ```

   Important: Use the `-createTargetProfile` parameter when invoking WBIPostUpgrade. This option creates a matching required new target profile for migration. For more information about target profiles, refer to "Target profile considerations" on page 9.

Note: i5/OS If you are migrating on an **i5/OS platform**, the target profile name must match the profile name of the source profile being migrated.

10. If required, manually update the databases used by WebSphere Process Server. Some database changes required by new versions of WebSphere Process Server are made automatically. However, if you have Business Process Choreographer or Business Space configured, or if the server has insufficient permissions to access the database schema, or other database-specific requirements are not met, you must update the database manually. For more information see "Upgrading databases for migration" on page 34.

Results

You have now migrated your configuration to an operating system that supports WebSphere Process Server.

What to do next

Verify that the migration has been successful. If your server has Business Process Choreographer configured, see "Postmigration tasks for Business Process Choreographer" on page 123. If your server has Business Space configured, see "Postmigration tasks for Business Space powered by WebSphere" on page 127. Finally, perform the checks described in "Postmigration tasks for WebSphere Process Server" on page 122.

Related concepts

Premigration considerations for Business Process Choreographer
If your servers run Business Process Choreographer, you should be aware of certain things that you need to plan and take into consideration before you migrate Business Process Choreographer.

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

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

Related tasks

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

Postmigration tasks for WebSphere Process Server
After migration, you should check some configuration settings. You might need to change them, or further configure the version 6.2 server.

Migrating to a remote system
Use the migration tools to migrate from an older version on one system to a newer version of WebSphere Process Server on a different, remote system. (Stand-alone servers only.)

Creating profiles
Learn how to create new WebSphere Enterprise Service Bus or WebSphere Process Server profiles. You can create profiles from a command line by using the manageprofiles command, or interactively by using the Profile Management Tool graphical user interface (GUI).

Related reference

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

WBIPreUpgrade command-line utility
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-line utility
Use the WBIPostUpgrade command for WebSphere Process Server to retrieve the profile configuration that was saved by the WBIPreUpgrade command at the *backupDirectory* that you had specified.

Related information

Installing and configuring WebSphere Process Server
This section describes how to prepare for, install, and configure an installation of IBM WebSphere Process Server. Instructions are provided for Linux, i5/OS, UNIX, and Windows systems.

Migrating a network deployment environment

Migrate a WebSphere Process Server network deployment environment.

Before you begin

See “Overview of migrating” on page 1 and “Premigration considerations for WebSphere Process Server” on page 4.

About this task

To migrate a network deployment environment, you need to first migrate the deployment manager, and then migrate its managed nodes.

Select the appropriate migration scenario for information on how to migrate to a WebSphere Process Server version 6.2 deployment cell. For managed nodes that are not in a clustered environment, see “Migrating non-clustered managed nodes” on page 81. For managed nodes in a clustered environment, see “Migrating clusters” on page 94.

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 114 before starting to migrate.

Tip: For help in troubleshooting problems when migrating, see “Troubleshooting version-to-version migration” on page 140.

Migrating a deployment manager

Migrate a WebSphere Process Server deployment manager, choosing from several methods depending on your needs.

Before you begin

See “Overview of migrating” on page 1 and “Premigration considerations for WebSphere Process Server” on page 4.

Select the appropriate migration scenario for information on how to migrate a WebSphere Process Server deployment manager from an older version to a newer version of WebSphere Process Server. (For example, a WebSphere Process Server Version 6.1 deployment manager to a WebSphere Process Server Version 6.2 deployment manager.)

- “Migrating a deployment manager using the migration wizard” on page 71

  This topic contains instructions for migrating an older version WebSphere Process Server deployment manager to a newer version deployment manager using the migration wizard (a graphical user interface).
- “Migrating a deployment manager using command-line tools” on page 76

  This topic contains instructions for migrating an older version WebSphere Process Server deployment manager to a newer version deployment manager using the migration command line tools rather than the migration wizard.

Tip: For help in troubleshooting problems when migrating, see “Troubleshooting version-to-version migration” on page 140.

Related concepts

Migrating clusters
Migrate clusters by migrating, in turn, each profile that contains cluster members following special procedures. Take additional steps if you want to minimize down time of cluster services.

Migrating a deployment manager using the migration wizard

Migrate a deployment manager from an older version to a newer version of WebSphere Process Server using the migration wizard.

Before you begin

Note: The migration wizard cannot run in a non-graphical environment. Examples of non-graphical environments include the **i5/OS platform** or telnet sessions. If you want to run migration in a non-graphical environment, use the WBIPreUpgrade and WBIPostUpgrade commands.

Note: The migration wizard supports only WebSphere Process Server profiles. If you have WebSphere Application Server profiles, you must use the migration commands.

Note: Migrate the WebSphere Process Server version 6.1.x or 6.0.2.x deployment manager to version 6.2 before migrating the managed nodes that comprise the cell. The deployment manager must always be at the highest release and fix level within a cell in order for it to manage all nodes in the cell. A version 6.2 deployment manager can manage version 6.1.x and version 6.2 managed nodes. For migrations from version 6.1.x, this allows a cell to be upgraded to a new release one node at a time, with minimal impact to the applications that are running within the cell. For migrations from version 6.0.2.x, to keep the cell running with minimal down time, you must migrate to version 6.2 in two stages: first from version 6.0.2.x to version 6.1.x, and then from version 6.1.x to version 6.2. Alternatively you can shut down the entire cell and migrate in a single step, directly from version 6.0.2.x to version 6.2.

Members of a cluster cannot be running different versions (6.0.2.x, 6.1.x, 6.2) of WebSphere Process Server. If you have configured a cluster containing servers running different versions, all the members running earlier versions of WebSphere Process Server must be stopped before you start the first version 6.2 cluster member. Also, once you start a version 6.2 cluster member, do not start any version 6.1.x or 6.0.2.x cluster members in that cluster.

Restrictions exist on using mixed-release cells. For more information see ″Restriction on using mixed-release cells″ found in the subtopics of Coexisting with other WebSphere product installations.

Make sure that the following conditions are met before you start the migration process:

- Your system meets all the hardware and software requirements for the new version of WebSphere Process Server.
- You have installed the new version of WebSphere Process Server side-by-side the previous version on the same system.
- A deployment manager profile, created with the older WebSphere Process Server version, resides on the same system.

- Sufficient disk space is available for the migrated profile and its backup. See “Premigration considerations for WebSphere Process Server” on page 4 for details about disk space requirements.

Make sure that you have completed the following tasks before you start the migration process:

- Back up the databases that support version 6.0.2.x or version 6.1.x WebSphere Process Server components.

Procedure

1. Log on as the root user on a Linux or UNIX system, or as a member of the Administrator group on a Windows system.
2. Identify, in advance, the pre-existing information required by the migration wizard, as listed below:

   Cell name
   Name of the cell managed by the deployment manager that you are migrating. The new version cell name must match the name in the old version’s configuration.

   Node name
   Name of the node that you are migrating. The new version node name must match the name in the old version’s configuration.

   Installation root directory
   See WBIPreUpgrade command-line utility for a description of the currentWebSphereDirectory parameter.

   Migration backup directory name
   See WBIPreUpgrade command-line utility for a description of the backupDirectory parameter.

   Administrative security user name (required if adminstrative security is configured)
   See WBIPostUpgrade command-line utility for a description of the -username parameter.

   Administrative security password (required if adminstrative security is configured)
   See WBIPostUpgrade command-line utility for a description of the -password parameter.

   Source profile name
   See WBIPostUpgrade command-line utility for a description of the -oldProfile parameter.

   Target profile name
   See WBIPostUpgrade command-line utility for a description of the -profileName parameter.

   Port value assignments (optional)
   See WBIPostUpgrade command-line utility for a description of the -replacePorts and -portBlock parameters.

   Note: This applies only if you are migrating from version 6.0.2.x toversion 6.2.
3. Optional: Create a new version 6.2 profile as the migration target using the **Create new profile** option in the migration wizard. For more information about creating target profiles, refer to “Target profile considerations” on page 9.

4. Stop the deployment manager that you are about to migrate. Use the stopManager command from the deployment manager's *profile_dir*/bin directory or from the deployment manager's First steps console.

   Use the following syntax:

   - Linux UNIX **On Linux and UNIX platforms:** `profile_root/bin/stopManager.sh`
   - Windows **On Windows platforms:** `profile_root\bin\stopManager.bat`

   If you have security enabled, specify the -username and -password parameters of the command. For more information about the stopManager command see the stopManager command.

   Note: You must stop the previous version deployment manager before you start the migration process. If you do not, any configuration changes you make during migration process will not be migrated to the target profile.

5. Invoke the migration wizard from the WebSphere Process Server to which you are migrating.

   Invoke the migration wizard in one of the following ways:

   - From the WebSphere Process Server First Steps console, select **Migration wizard**.
   - Run one of the following scripts (depending upon your operating system) stored in the *install_dir*/bin directory:
     - Linux UNIX **On Linux and UNIX platforms:** `wbi_migration.sh`
     - Windows **On Windows platforms:** `wbi_migration.bat`

     Note: You can optionally change the default trace setting (`*=all=enabled:com.ibm.ws.migration.common.*=all=disabled`) when invoking the migration wizard. The default trace setting enables tracing on only certain classes, but you can change the default to either enable full tracing or disable all tracing.

     - To enable full tracing, run one of the following scripts to invoke the migration wizard, depending on your operating system:
       - Linux UNIX **On Linux and UNIX platforms:** `wbi_migration.sh -W -migrationPanel.traceString="*=all=enabled"`
       - Windows **On Windows platforms:** `wbi_migration.bat -W -migrationPanel.traceString="*=all=enabled"`
     - To disable all tracing, run one of the following scripts to invoke the migration wizard, depending on your operating system:
       - Linux UNIX **On Linux and UNIX platforms:** `wbi_migration.sh -W -migrationPanel.traceString="*=all=disabled"`
       - Windows **On Windows platforms:** `wbi_migration.bat -W -migrationPanel.traceString="*=all=disabled"`

   For information about what processing the migration wizard actually performs, see "What the migration wizard does" on page 19.

6. Follow the prompts for the migration wizard as described in "Running the migration wizard" on page 21.

7. If you need to manually update the Common database, do so now. See "Upgrading the Common database manually" on page 35 for instructions. Normally, database changes required by new versions of WebSphere Process Server are made automatically. When the server is first started the database

tables are migrated to the new schema version. However, in cases in which the server has insufficient permissions to access the database schema, or other database-specific requirements are not met, you must update the database manually.

Note: If a server or cluster has Business Process Choreographer configured, upgrade the database after you migrate the server or cluster, not when you migrate the deployment manager.

8. Start the version 6.2 deployment manager.

   CAUTION:
   When the version 6.2 deployment manager is started, the federated nodes synchronize with the migrated deployment manager. This synchronization causes the applications to be reinstalled. If you have any applications running on active servers, those applications will appear to restart and be unavailable for a very short time.
   To start the deployment manager, use the startManager command from the *profile_dir*/bin directory or the First steps console. See startManager command for more information on the startManager command.

9. Optional: Uninstall the version 6.1.x or 6.0.2.x deployment manager.

   Perform this step only after you are certain that you have successfully migrated the configuration of the deployment manager that you intend to delete. For more information about uninstalling, see Uninstalling the software.

Results

Your deployment manager is now migrated.

What to do next

Verify that the migration has been successful. If a server or cluster being managed by this deployment manager has Business Process Choreographer configured, see "Premigration considerations for Business Process Choreographer" on page 12. Finally, perform the checks described in "Postmigration tasks for WebSphere Process Server" on page 122.

Related concepts

Premigration considerations for Business Process Choreographer
If your servers run Business Process Choreographer, you should be aware of certain things that you need to plan and take into consideration before you migrate Business Process Choreographer.

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

Migration wizard
The version-to-version migration wizard is a graphical interface that guides you through migrating from an older version to a newer version of WebSphere Process Server.

Related tasks

Running the migration wizard
Run the migration wizard on AIX, HP-UX, Linux, Solaris, or Windows systems to migrate WebSphere Process Server.

Migrating a deployment manager using command-line tools
Migrate a deployment manager from an older version to a newer version of WebSphere Process Server using the command-line tools.

Migrating non-clustered managed nodes
Migrate a WebSphere Process Server managed node, choosing from several methods depending on your needs.

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

Postmigration tasks for WebSphere Process Server
After migration, you should check some configuration settings. You might need to change them, or further configure the version 6.2 server.

Upgrading databases for migration
In conjunction with migration, the database schema of some WebSphere Process Server components must be upgraded. This can occur automatically but in some cases you must upgrade the schema manually.

Creating profiles
Learn how to create new WebSphere Enterprise Service Bus or WebSphere Process Server profiles. You can create profiles from a command line by using the manageprofiles command, or interactively by using the Profile Management Tool graphical user interface (GUI).

Rolling back a deployment cell
You can use the **restoreConfig** and **wsadmin** commands to roll back a migrated WebSphere Process Server version 6.2 deployment cell to version 6.1.x or 6.0.2.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

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

WBIPostUpgrade command-line utility
Use the WBIPostUpgrade command for WebSphere Process Server to retrieve the profile configuration that was saved by the WBIPreUpgrade command at the *backupDirectory* that you had specified.

WBIPreUpgrade command-line utility
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

Uninstalling the software
Learn about the different ways of uninstalling IBM WebSphere Process Server.

Coexisting with other WebSphere product installations
An installation of WebSphere Process Server, version 6.2 can coexist on the same system with installations of any version of WebSphere Enterprise Service Bus, WebSphere Process Server, and with certain versions of selected WebSphere products.

Migrating a deployment manager using command-line tools

Migrate a deployment manager from an older version to a newer version of WebSphere Process Server using the command-line tools.

Before you begin

Note: In a WebSphere Process Server network deployment cell, the deployment manager must always be in a WebSphere Process Server profile.

Note: Migrate the WebSphere Process Server version 6.1.x or 6.0.2.x deployment manager to version 6.2 before migrating the managed nodes that comprise the cell. The deployment manager must always be at the highest release and fix level within a cell in order for it to manage all nodes in the cell. A version 6.2 deployment manager can manage version 6.1.x and version 6.2 managed nodes. For migrations from version 6.1.x, this allows a cell to be upgraded to a new release one node at a time, with minimal impact to the applications that are running within the cell. For migrations from version 6.0.2.x, to keep the cell running with minimal down time, you must migrate to version 6.2 in two stages: first from version 6.0.2.x to version 6.1.x, and then from version 6.1.x to version 6.2. Alternatively you can shut down the entire cell and migrate in a single step, directly from version 6.0.2.x to version 6.2.

Members of a cluster cannot be running different versions (6.0.2.x, 6.1.x, 6.2) of WebSphere Process Server. If you have configured a cluster containing servers running different versions, all the members running earlier versions of WebSphere Process Server must be stopped before you start the first version 6.2 cluster member. Also, once you start a version 6.2 cluster member, do not start any version 6.1.x or 6.0.2.x cluster members in that cluster.

Make sure that the following conditions are met before you start the migration process:

- Your system meets all the hardware and software requirements for the new version of WebSphere Process Server.
- You have installed the new version of WebSphere Process Server side-by-side the previous version on the same system.

- A deployment manager profile, created with the older WebSphere Process Server version, resides on the same system.
- Sufficient disk space is available for the migrated profile and its backup. See "Premigration considerations for WebSphere Process Server" on page 4 for details about disk space requirements.

Make sure that you have completed the following tasks before you start the migration process:

- Back up the databases that support version 6.0.2.x or version 6.1.x WebSphere Process Server components.

Procedure

1. Log on using one of the following procedures, depending on your operating system.
   - i5/OS **On i5/OS platforms:** Log on with an i5/OS user profile that has *SECOFR user class or *ALLOBJ special authority.
   - Linux UNIX **On Linux and UNIX platforms:** Log on as the root user.
   - Windows **On Windows platforms:** Log on as a member of the Administrator group.
2. Identify, in advance, the pre-existing information required by the migration wizard, as listed below:

   Cell name

   Name of the cell managed by the deployment manager that you are migrating. The new version cell name must match the name in the old version's configuration.

   Node name

   Name of the node that you are migrating. The new version node name must match the name in the old version's configuration.

   Installation root directory

   See WBIPreUpgrade command-line utility for a description of the currentWebSphereDirectory parameter.

   Migration backup directory name

   See WBIPreUpgrade command-line utility for a description of the backupDirectory parameter.

   Administrative security user name (required if adminstrative security is configured)

   See WBIPostUpgrade command-line utility for a description of the -username parameter.

   Administrative security password (required if adminstrative security is configured)

   See WBIPostUpgrade command-line utility for a description of the -password parameter.

   Source profile name

   See WBIPostUpgrade command-line utility for a description of the -oldProfile parameter.

   Target profile name

   See WBIPostUpgrade command-line utility for a description of the -profileName parameter.

Port value assignments (optional)
: See WBIPostUpgrade command-line utility for a description of the -replacePorts and -portBlock parameters.

Note: This applies only if you are migrating from version 6.0.2.x toversion 6.2.

3. Stop the deployment manager that you are about to migrate. Use the stopManager command from the deployment manager's *profile_dir*/bin directory or from the deployment manager's First steps console.

   For more information about the stopServer command see the stopServer command. Use the following syntax:

   Note: **On i5/OS platforms**, you must run the scripts under QSHELL. To start a QSHELL session, open a CL command prompt and type QSH.

   - i5/OS **On i5/OS platforms:** *profile_root*/bin/stopManager
   - Linux UNIX **On Linux and UNIX platforms:** *profile_root*/bin/stopManager.sh
   - Windows **On Windows platforms:** *profile_root*\bin\stopManager.bat

   If security is enabled, use one of the following commands instead. The user name provided must be a member of the operator or administrator role.

   - i5/OS **On i5/OS platforms:** *profile_root*/bin/stopManager -username *user_ID* -password *password*
   - Linux UNIX **On Linux and UNIX platforms:** *profile_root*/bin/stopManager.sh -username *user_ID* -password *password*
   - Windows **On Windows platforms:** *profile_root*\bin\stopManager.bat -username *user_ID* -password *password*

   Note: You must stop the previous version deployment manager before you start the migration process. If you do not, any configuration changes you make during migration process will not be migrated to the target profile.
4. Run the WBIPreUpgrade command, specifying the migration backup directory name and the existing WebSphere Process Server directory name. The WBIPreUpgrade tool saves selected files from the *install_root* and *profile_root* directories to a backup directory you specify. See the WBIPreUpgrade command-line utility for details.
5. Run the WBIPostUpgrade command, specifying the migration backup directory. The WBIPostUpgrade tool restores the backed up configuration in the backup directory into the new WebSphere Process Server Deployment Manager profile. See the WBIPostUpgrade command-line utility for details.

   Important: Use the -createTargetProfile parameter when invoking WBIPostUpgrade. This option creates a matching required new target profile for migration. For more information about target profiles, refer to "Target profile considerations" on page 9.

   Note: i5/OS If you are migrating on an **i5/OS platform**, the target profile name must match the profile name of the source profile being migrated.
6. If you need to manually update the Common database, do so now. See "Upgrading the Common database manually" on page 35 for instructions. Normally, database changes required by new versions of WebSphere Process Server are made automatically. When the server is first started the database tables are migrated to the new schema version. However, in cases in which the

server has insufficient permissions to access the database schema, or other database-specific requirements are not met, you must update the database manually.

Note: If a server or cluster has Business Process Choreographer configured, upgrade the database after you migrate the server or cluster, not when you migrate the deployment manager.

7. Start the version 6.2 deployment manager.

   CAUTION:
   When the version 6.2 deployment manager is started, the federated nodes synchronize with the migrated deployment manager. This synchronization causes the applications to be reinstalled. If you have any applications running on active servers, those applications will appear to restart and be unavailable for a very short time.
   To start the deployment manager, use the startManager command from the *profile_dir*/bin directory or the First steps console. See startManager command for more information on the startManager command.
8. Optional: Uninstall the version 6.1.x or 6.0.2.x deployment manager.

   Perform this step only after you are certain that you have successfully migrated the configuration of the deployment manager that you intend to delete. For more information about uninstalling, see Uninstalling the software.

Results

Your deployment manager is now migrated.

What to do next

Verify that the migration has been successful. If a server or cluster being managed by this deployment manager has Business Process Choreographer configured, see “Premigration considerations for Business Process Choreographer” on page 12. Finally, perform the checks described in “Postmigration tasks for WebSphere Process Server” on page 122.

Related concepts

Premigration considerations for Business Process Choreographer
If your servers run Business Process Choreographer, you should be aware of certain things that you need to plan and take into consideration before you migrate Business Process Choreographer.

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

Migration wizard
The version-to-version migration wizard is a graphical interface that guides you through migrating from an older version to a newer version of WebSphere Process Server.

Related tasks

Migrating a deployment manager using the migration wizard
Migrate a deployment manager from an older version to a newer version of WebSphere Process Server using the migration wizard.

Migrating non-clustered managed nodes
Migrate a WebSphere Process Server managed node, choosing from several methods depending on your needs.

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

Postmigration tasks for WebSphere Process Server
After migration, you should check some configuration settings. You might need to change them, or further configure the version 6.2 server.

Upgrading databases for migration
In conjunction with migration, the database schema of some WebSphere Process Server components must be upgraded. This can occur automatically but in some cases you must upgrade the schema manually.

Creating profiles
Learn how to create new WebSphere Enterprise Service Bus or WebSphere Process Server profiles. You can create profiles from a command line by using the manageprofiles command, or interactively by using the Profile Management Tool graphical user interface (GUI).

Rolling back a deployment cell
You can use the **restoreConfig** and **wsadmin** commands to roll back a migrated WebSphere Process Server version 6.2 deployment cell to version 6.1.x or 6.0.2.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

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

WBIPostUpgrade command-line utility
Use the WBIPostUpgrade command for WebSphere Process Server to retrieve the profile configuration that was saved by the WBIPreUpgrade command at the *backupDirectory* that you had specified.

WBIPreUpgrade command-line utility
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

Uninstalling the software
Learn about the different ways of uninstalling IBM WebSphere Process Server.

Coexisting with other WebSphere product installations
An installation of WebSphere Process Server, version 6.2 can coexist on the same system with installations of any version of WebSphere Enterprise Service Bus, WebSphere Process Server, and with certain versions of selected WebSphere products.

Migrating non-clustered managed nodes

Migrate a WebSphere Process Server managed node, choosing from several methods depending on your needs.

Before you begin

See "Overview of migrating" on page 1 and "Premigration considerations for WebSphere Process Server" on page 4.

Select the appropriate subtopic for information on how to migrate a WebSphere Process Server non-clustered managed node from an older version to a newer version of WebSphere Process Server.

Note: If your managed nodes are part of a cluster, follow the instructions in "Migrating clusters" on page 94.

Tip: For help in troubleshooting problems when migrating, see "Troubleshooting version-to-version migration" on page 140.

Related tasks

Migrating a cluster
To migrate a cluster, migrate each profile containing a member of that cluster one at a time. The migration requires extra steps not required for a non-clustered environment.

Migrating a cluster with minimal down time
To migrate a cluster while minimizing down time, first migrate approximately half of the profiles contributing to the cluster, then migrate the other half. Perform the extra steps required for cluster migration after you have migrated the first set of profiles.

Migrating a deployment manager using the migration wizard
Migrate a deployment manager from an older version to a newer version of WebSphere Process Server using the migration wizard.

Migrating a deployment manager using command-line tools
Migrate a deployment manager from an older version to a newer version of WebSphere Process Server using the command-line tools.

Migrating non-clustered managed nodes using the migration wizard

Migrate non-clustered managed nodes from an older version to a newer version of WebSphere Process Server using the migration wizard.

Before you begin

Note: The migration wizard cannot run in a non-graphical environment. Examples of non-graphical environments include the **i5/OS platform** or telnet sessions. If you want to run migration in a non-graphical environment, use the WBIPreUpgrade and WBIPostUpgrade commands.

Note: The migration wizard supports only WebSphere Process Server profiles. If you have WebSphere Application Server profiles, you must use the migration commands.

Make sure that the following conditions are met before you start the migration process:

- Your system meets all the hardware and software requirements for the new version of WebSphere Process Server.
- You have installed the new version of WebSphere Process Server side-by-side the previous version on the same system.
- A federated profile, created with the older WebSphere Process Server version resides on the same system.
- Sufficient disk space is available for the migrated profile and its backup. See "Premigration considerations for WebSphere Process Server" on page 4 for disk space requirements.
- The deployment manager that manages the managed node that you intend to migrate has already been migrated to the newer version of WebSphere Process Server, and is running.

  Note: Migrating a WebSphere Process Server version 6.1.x or 6.0.2.x managed node to a version 6.2 managed node requires that you first migrate the version 6.1.x or 6.0.2.x deployment manager to a version 6.2 deployment manager. See "Migrating a deployment manager" on page 70 for instructions. Complete the migration of the deployment manager before you proceed with the instructions in this topic.

Make sure that you have completed the following tasks before you start the migration process:

- Back up the databases that support version 6.1.x or 6.0.2.x WebSphere Process Server components.

About this task

After you migrate an older version deployment manager to a newer version of WebSphere Process Server, the newer version deployment manager runs in compatibility mode by default, where it can manage both older and newer versions of WebSphere Process Server. For example, after migration, a version 6.2 deployment manager can manage both version 6.1.x and version 6.2 nodes. The managed nodes of the previous version 6.1.x deployment manager are now running as version 6.1.x managed nodes in the version 6.2 deployment manager.

Over time, migrate eachversion 6.1.x WebSphere Process Server managed node (server managed by a version 6.2 deployment manager) to a version 6.2 managed node. After migrating all theversion 6.1.x managed nodes, use the **convertScriptCompatibility** script to change the deployment manager from supporting compatibility of version 6.1.x administration scripts to supporting compatibility of only version 6.1.x and version 6.2 administration scripts. See the convertScriptCompatibility command.

Note: When following the directions at this link to use the convertScriptCompatibility command, use the WBIPostUpgrade command rather than the WASPostUpgrade command.

For help in troubleshooting problems when migrating, see “Troubleshooting version-to-version migration” on page 140.

Procedure

1. Log on as the root user on a Linux or UNIX system, or as a member of the Administrator group on a Windows system.
2. Stop the version 6.1.x or version 6.0.2.x server if it is running on the node to be migrated. Use the stopServer command from the *profile_dir*/bin directory for the profile of the affected server, or stop the server from the profile’s First steps console.

   For more information about the stopServer command see the stopServer command. Use the following syntax:

   - Linux UNIX **On Linux and UNIX platforms:** `profile_root/bin/stopServer.sh server_name`
   - Windows **On Windows platforms:** `profile_root\bin\stopServer.bat server_name`

   If security is enabled, use one of the following commands instead. The user name provided must be a member of the operator or administrator role.

   - Linux UNIX **On Linux and UNIX platforms:** `profile_root/bin/stopServer.sh server_name -username user_ID -password password`
   - Windows **On Windows platforms:** `profile_root\bin\stopServer.bat server_name -username user_ID -password password`

   On the Windows operating system, even if security is enabled, the -username and -password parameters do not have to be specified if the server is running as a Windows service. In this case, the parameters are automatically passed into the script that the Windows service uses to shut down the system.

Note: Before you start the migration process, you must stop the server from which you are migrating. It is not necessary to have that server running in order to migrate its configuration. The migration tools can retrieve all of the configuration data while the server is stopped.

3. Stop the node agent of the node to be migrated. Issue one of the following commands to stop the nodeagent process, depending on platform (where *profile_root* represents the installation directory of the federated node):

   - Linux UNIX **On Linux and UNIX platforms:** `profile_root/bin/stopNode.sh`
   - Windows **On Windows platforms:** `profile_root\bin\stopNode.bat`

   If security is enabled, use one of the following commands instead:

   - Linux UNIX **On Linux and UNIX platforms:** `profile_root/bin/stopNode.sh -username user_ID -password password`
   - Windows **On Windows platforms:** `profile_root\bin\stopNode.bat -username user_ID -password password`

4. Identify, in advance, the pre-existing information required for the migration, as listed below:

   Installation root directory

   See WBIPreUpgrade command-line utility for a description of the currentWebSphereDirectory parameter.

   Migration backup directory name

   See WBIPreUpgrade command-line utility for a description of the backupDirectory parameter.

   Administrative security user name (required if adminstrative security is configured)

   See WBIPostUpgrade command-line utility for a description of the -username parameter.

   Administrative security password (required if adminstrative security is configured)

   See WBIPostUpgrade command-line utility for a description of the -password parameter.

   Source profile name

   See WBIPostUpgrade command-line utility for a description of the -oldProfile parameter.

   Target profile name

   See WBIPostUpgrade command-line utility for a description of the -profileName parameter.

   Port value assignments (optional)

   See WBIPostUpgrade command-line utility for a description of the -replacePorts and -portBlock parameters.

   Note: This applies only if you are migrating from version 6.0.2.x toversion 6.2.

5. Ensure that the version 6.2 deployment manager is up and running.
6. Invoke the migration wizard.

   Invoke the migration wizard in one of the following ways:

- From the WebSphere Process Server First Steps console, select **Migration wizard**.
- Run one of the following scripts (depending upon your operating system) stored in the *install_dir*/bin directory:
  - Linux UNIX **On Linux and UNIX platforms:** `wbi_migration.sh`
  - Windows **On Windows platforms:** `wbi_migration.bat`

  Note: You can optionally change the default trace setting (`*=all=enabled:com.ibm.ws.migration.common.*=all=disabled`) when invoking the migration wizard. The default trace setting enables tracing on only certain classes, but you can change the default to either enable full tracing or disable all tracing.
  - To enable full tracing, run one of the following scripts to invoke the migration wizard, depending on your operating system:
    - Linux UNIX **On Linux and UNIX platforms:** `wbi_migration.sh -W -migrationPanel.traceString="*=all=enabled"`
    - Windows **On Windows platforms:** `wbi_migration.bat -W -migrationPanel.traceString="*=all=enabled"`
  - To disable all tracing, run one of the following scripts to invoke the migration wizard, depending on your operating system:
    - Linux UNIX **On Linux and UNIX platforms:** `wbi_migration.sh -W -migrationPanel.traceString="*=all=disabled"`
    - Windows **On Windows platforms:** `wbi_migration.bat -W -migrationPanel.traceString="*=all=disabled"`

The migration wizard copies the configuration and applications from the version 6.1.x or 6.0.2.x managed node to the version 6.2 managed node. After migrating all of the data, the wizard federates the version 6.2 managed node into the deployment manager cell.

7. Stop (if they are not stopped already) the server and node agent. If the server has not already been stopped, stop the server as described in step 2. If the node agent has not already been stopped, stop the node agent as described in step 3.
8. If you are migrating a server that has Business Process Choreographer or Business Space configured, you must upgrade the associated databases manually at this point. See “Upgrading the Business Process Choreographer database manually” on page 38 or “Upgrading the Business Space database manually” on page 47.
9. Restart the node agent. To start a node agent, run the command *profile_root*\bin\startNode (where *profile_root* represents the installation directory of the managed node).
   - Linux UNIX **On Linux and UNIX platforms:** `*profile_root*/bin/startNode.sh`
   - Windows **On Windows platforms:** `*profile_root*\bin\startNode.bat`
10. Start the server or servers running on this node. Start each server using the startServer command, the administrative console, or the profile’s First steps console. For more information see Starting an application server.
11. Repeat steps 1-10 for each additional managed node you wish to migrate.

12. If you chose the compatibility option (which is the default), and if all of your nodes are completely migrated to WebSphere Process Server version 6.2, run the convertScriptCompatibility script to remove compatibility from the version 6.2 deployment manager.

    Note: This applies only if you are migrating from version 6.0.2.x.

    Issue the convertScriptCompatibility command from the bin directory.

    - UNIX Linux *install_root*/bin/convertScriptCompatibility.sh
    - Windows *install_root*\bin\convertScriptCompatibility.bat

    See the convertScriptCompatibility command.

Results

Your non-clustered managed nodes are now migrated.

What to do next

Verify that the migration has been successful. If your server has Business Process Choreographer configured, see "Postmigration tasks for Business Process Choreographer" on page 123. If your server has Business Space configured, see "Postmigration tasks for Business Space powered by WebSphere" on page 127. Finally, perform the checks described in "Postmigration tasks for WebSphere Process Server" on page 122.

Related concepts

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

Migration wizard
The version-to-version migration wizard is a graphical interface that guides you through migrating from an older version to a newer version of WebSphere Process Server.

Related tasks

Running the migration wizard
Run the migration wizard on AIX, HP-UX, Linux, Solaris, or Windows systems to migrate WebSphere Process Server.

Migrating non-clustered managed nodes using the command-line tools
Migrate non-clustered managed nodes from an older version to a newer version of WebSphere Process Server with the command-line tools.

Stopping managed servers
Stop the server before modifying certain configurations for the server or the modules deployed on it. Note that you can stop a single server in a cluster.

Upgrading the Business Process Choreographer database manually
After migrating a server or cluster that has Business Process Choreographer configured, you must manually upgrade the schema for the Business Process Choreographer database and perform a data migration before you start the server or any cluster member.

Upgrading databases for migration
In conjunction with migration, the database schema of some WebSphere Process Server components must be upgraded. This can occur automatically but in some cases you must upgrade the schema manually.

Creating profiles
Learn how to create new WebSphere Enterprise Service Bus or WebSphere Process Server profiles. You can create profiles from a command line by using the manageprofiles command, or interactively by using the Profile Management Tool graphical user interface (GUI).

Rolling back a managed node
You can use the **restoreConfig** and **wsadmin** commands to roll back a migrated WebSphere Process Server version 6.2 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 tasks for WebSphere Process Server
After migration, you should check some configuration settings. You might need to change them, or further configure the version 6.2 server.

Related reference

WBIPostUpgrade command-line utility
Use the WBIPostUpgrade command for WebSphere Process Server to retrieve the profile configuration that was saved by the WBIPreUpgrade command at the *backupDirectory* that you had specified.

WBIPreUpgrade command-line utility
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

Starting an application server

convertScriptCompatibility command

startManager command

backupConfig command

stopServer command

Migrating non-clustered managed nodes using the command-line tools

Migrate non-clustered managed nodes from an older version to a newer version of WebSphere Process Server with the command-line tools.

Before you begin

Note: When you migrate using the command-line tools, you can migrate either a WebSphere Process Server profile or a WebSphere Application Server profile.

Make sure that the following conditions are met before you start the migration process:

- Your system meets all the hardware and software requirements for the new version of WebSphere Process Server.
- You have installed the new version of WebSphere Process Server side-by-side the previous version on the same system.
- A federated profile, created with the older WebSphere Process Server version resides on the same system.
- Sufficient disk space is available for the migrated profile and its backup. See "Premigration considerations for WebSphere Process Server" on page 4 for disk space requirements.
- The deployment manager that manages the managed node that you intend to migrate has already been migrated to the newer version of WebSphere Process Server, and is running.

  Note: Migrating a WebSphere Process Server version 6.1.x or 6.0.2.x managed node to a version 6.2 managed node requires that you first migrate the version 6.1.x or 6.0.2.x deployment manager to a version 6.2 deployment manager. See "Migrating a deployment manager" on page 70 for instructions. Complete the migration of the deployment manager before you proceed with the instructions in this topic.

Make sure that you have completed the following tasks before you start the migration process:

- Back up the databases that support version 6.1.x or 6.0.2.x WebSphere Process Server components.

About this task

After you migrate a version 6.0.2.x deployment manager to a newer version of WebSphere Process Server, the newer version deployment manager runs in compatibility mode by default, where it can manage both older and newer versions of WebSphere Process Server. For example, after migration, a version 6.2

deployment manager can manage both version 6.0.2.x and version 6.2 nodes. In other words, version 6.0.2.x managed nodes can run with the version 6.2 deployment manager. Over time, you can migrate each version 6.0.2.x WebSphere Process Server managed node (server managed by a version 6.2 deployment manager) to a version 6.2 managed node. After migrating all version 6.0.2.x managed nodes, you use the `convertScriptCompatibility` script to convert their configurations from a mode that supports compatibility of version 6.0.2.x administration scripts to a mode that is fully in a configuration version 6.2 model. See the convertScriptCompatibility command.

Note: When following the directions at this link to use the convertScriptCompatibility command, use the WBIPostUpgrade command rather than the WASPostUpgrade command.

Procedure

1. Log on using one of the following procedures, depending on your operating system.
   - i5/OS **On i5/OS platforms:** Log on with an i5/OS user profile that has `*SECOFR` user class or `*ALLOBJ` special authority.
   - Linux UNIX **On Linux and UNIX platforms:** Log on as the root user.
   - Windows **On Windows platforms:** Log on as a member of the Administrator group.
2. Stop the version 6.1.x or 6.0.2.x server if it is running on the node to be migrated. Use the stopServer command from the *profile_dir*/bin directory for the profile of the affected server, or stop the server from the profile's First steps console.

   For more information about the stopServer command see the stopServer command. Use the following syntax:

   Note: On i5/OS platforms, you must run the scripts under QSHELL. To start a QSHELL session, open a CL command prompt and type `QSH`.
   - i5/OS **On i5/OS platforms:** *profile_root*/bin/stopServer *server_name*
   - Linux UNIX **On Linux and UNIX platforms:** *profile_root*/bin/stopServer.sh *server_name*
   - Windows **On Windows platforms:** *profile_root*\bin\stopServer.bat *server_name*

   If security is enabled, use one of the following commands instead. The user name provided must be a member of the operator or administrator role.
   - i5/OS **On i5/OS platforms:** *profile_root*/bin/stopServer *server_name* -username *user_ID* -password *password*
   - Linux UNIX **On Linux and UNIX platforms:** *profile_root*/bin/stopServer.sh *server_name* -username *user_ID* -password *password*
   - Windows **On Windows platforms:** *profile_root*\bin\stopServer.bat *server_name* -username *user_ID* -password *password*

   On the Windows operating system, even if security is enabled, the -username and -password parameters do not have to be specified if the server is running as a Windows service. In this case, the parameters are automatically passed into the script that the Windows service uses to shut down the system.

Note: Stop the server before beginning the migrating process. By default, all servers on the node are stopped before the migration completes.

3. Stop the node agent of the node to be migrated. Issue one of the following commands to stop the nodeagent process, depending on platform (where *profile_root* represents the installation directory of the federated node):

   - i5/OS **On i5/OS platforms:** *profile_root*/bin/stopNode
   - Linux UNIX **On Linux and UNIX platforms:** *profile_root*/bin/stopNode.sh
   - Windows **On Windows platforms:** *profile_root*\bin\stopNode.bat

   If security is enabled, use one of the following commands instead:

   - i5/OS **On i5/OS platforms:** *profile_root*/bin/stopNode -username *user_ID* -password *password*
   - Linux UNIX **On Linux and UNIX platforms:** *profile_root*/bin/stopNode.sh -username *user_ID* -password *password*
   - Windows **On Windows platforms:** *profile_root*\bin\stopNode.bat -username *user_ID* -password *password*

   Note: You must stop the old node before you start migration process. It is not necessary to have the server running to migrate its configuration. The migration tools can retrieve all the configuration data while the server is stopped.

4. Identify, in advance, the pre-existing information required for the migration, as listed below:

   Installation root directory
   See WBIPreUpgrade command-line utility for a description of the currentWebSphereDirectory parameter.

   Migration backup directory name
   See WBIPreUpgrade command-line utility for a description of the backupDirectory parameter.

   Administrative security user name (required if adminstrative security is configured)
   See WBIPostUpgrade command-line utility for a description of the -username parameter.

   Administrative security password (required if adminstrative security is configured)
   See WBIPostUpgrade command-line utility for a description of the -password parameter.

   Source profile name
   See WBIPostUpgrade command-line utility for a description of the -oldProfile parameter.

   Target profile name
   See WBIPostUpgrade command-line utility for a description of the -profileName parameter.

   Port value assignments (optional)
   See WBIPostUpgrade command-line utility for a description of the -replacePorts and -portBlock parameters.

   Note: This applies only if you are migrating from version 6.0.2.x toversion 6.2.

5. Ensure that the version 6.2 deployment manager is up and running.
6. Run the WBIPreUpgrade command, specifying the migration backup directory name and the existing WebSphere Process Server directory name. The WBIPreUpgrade tool saves the configuration files of your existing profiles to the backup directory you specify.
7. Run the WBIPostUpgrade command, specifying the migration backup directory. The WBIPostUpgrade tool restores the backed up configuration in the backup directory into the new WebSphere Process Server Deployment Manager profile.

   Important: Use the `-createTargetProfile` parameter when invoking WBIPostUpgrade. This option creates a matching required new target profile for migration. For more information about target profiles, refer to "Target profile considerations" on page 9.

   Note: i5/OS If you are migrating on an **i5/OS platform**, the target profile name must match the profile name of the source profile being migrated.
8. Stop (if they are not stopped already) the server and node agent. If the server has not already been stopped, stop the server as described in step 2. If the node agent has not already been stopped, stop the node agent as described in step 3.
9. If you are migrating a server that has Business Process Choreographer or Business Space configured, you must upgrade the associated databases manually at this point. See "Upgrading the Business Process Choreographer database manually" on page 38 or "Upgrading the Business Space database manually" on page 47.
10. Restart the node agent. To start a node agent, run the command *profile_root*\bin\startNode (where *profile_root* represents the installation directory of the managed node).
    - i5/OS **On i5/OS platforms:** `*profile_root*/bin/startNode`
    - Linux UNIX **On Linux and UNIX platforms:** `*profile_root*/bin/startNode.sh`
    - Windows **On Windows platforms:** `*profile_root*\bin\startNode.bat`
11. Start the server or servers running on this node. Start each server using the startServer command, the administrative console, or the profile's First steps console. For more information see Starting an application server.
12. Repeat steps 1- 11 for each additional managed node you wish to migrate.

    Note: You need to perform step 6 (running WBIPreUpgrade) again only you are migrating from version 6.1.x, or if you are migrating from version 6.0.2.x and the version 6.0.2.x system has been reconfigured since the first time you ran WBIPreUpgrade.
13. If you chose the compatibility option (which is the default), and if all of your nodes are completely migrated to WebSphere Process Server version 6.2, run the convertScriptCompatibility script to remove compatibility from the version 6.2 deployment manager.

    Note: This applies only if you are migrating from version 6.0.2.x.

    Issue one of the convertScriptCompatibility commands from the bin directory, depending on your operating system:
    - i5/OS **On i5/OS platforms:** *install_root*/bin/convertScriptCompatibility

- Linux UNIX **On Linux/UNIX platforms:** *install_root*/bin/ convertScriptCompatibility.sh
- Windows **On Windows platforms:** *install_root*\bin\ convertScriptCompatibility.bat

See the convertScriptCompatibility command.

Results

Your non-clustered managed nodes are now migrated.

What to do next

Verify that the migration has been successful. If your server has Business Process Choreographer configured, see “Postmigration tasks for Business Process Choreographer” on page 123. If your server has Business Space configured, see “Postmigration tasks for Business Space powered by WebSphere” on page 127. Finally, perform the checks described in “Postmigration tasks for WebSphere Process Server” on page 122.

Related concepts

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

Migration wizard
The version-to-version migration wizard is a graphical interface that guides you through migrating from an older version to a newer version of WebSphere Process Server.

Related tasks

Migrating non-clustered managed nodes using the migration wizard
Migrate non-clustered managed nodes from an older version to a newer version of WebSphere Process Server using the migration wizard.

Stopping managed servers
Stop the server before modifying certain configurations for the server or the modules deployed on it. Note that you can stop a single server in a cluster.

Upgrading the Business Process Choreographer database manually
After migrating a server or cluster that has Business Process Choreographer configured, you must manually upgrade the schema for the Business Process Choreographer database and perform a data migration before you start the server or any cluster member.

Upgrading databases for migration
In conjunction with migration, the database schema of some WebSphere Process Server components must be upgraded. This can occur automatically but in some cases you must upgrade the schema manually.

Creating profiles
Learn how to create new WebSphere Enterprise Service Bus or WebSphere Process Server profiles. You can create profiles from a command line by using the manageprofiles command, or interactively by using the Profile Management Tool graphical user interface (GUI).

Rolling back a managed node
You can use the **restoreConfig** and **wsadmin** commands to roll back a migrated WebSphere Process Server version 6.2 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 tasks for WebSphere Process Server
After migration, you should check some configuration settings. You might need to change them, or further configure the version 6.2 server.

Related reference

WBIPostUpgrade command-line utility
Use the WBIPostUpgrade command for WebSphere Process Server to retrieve the profile configuration that was saved by the WBIPreUpgrade command at the *backupDirectory* that you had specified.

WBIPreUpgrade command-line utility
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

Starting an application server

convertScriptCompatibility command

startManager command

backupConfig command

stopServer command

Migrating clusters

Migrate clusters by migrating, in turn, each profile that contains cluster members following special procedures. Take additional steps if you want to minimize down time of cluster services.

Migrating a cluster requires you to stop all node agents and servers that contribute to a cluster before migrating each profile. In addition, you must run the WBIProfileUpgrade script for each cluster. Choose from the following subtopics, depending on your needs:

Related tasks

Migrating a deployment manager
Migrate a WebSphere Process Server deployment manager, choosing from several methods depending on your needs.

Migrating a cluster

To migrate a cluster, migrate each profile containing a member of that cluster one at a time. The migration requires extra steps not required for a non-clustered environment.

Before you begin

You must have an existing cell containing at least one cluster running on an older version of WebSphere Process Server (for example, version 6.1.x or 6.0.2.x) that you wish to migrate to a newer version (for example, version 6.2). In addition, you must have installed the new version of WebSphere Process Server.

Important: In a cluster, version 6.1.x or 6.0.2.x members and version 6.2 members must never run at the same time. All version 6.1.x or 6.0.2.x cluster members must be stopped before you start the first version 6.2 cluster member. Also, once you start a version 6.2 cluster member, do not start any version 6.1.x or 6.0.2.x cluster members in that cluster.

About this task

Following these steps will ensure that you retain cluster functionality on the new version of WebSphere Process Server.

Procedure

1. Migrate the deployment manager. Follow one of the instruction sets listed in "Migrating a deployment manager" on page 70 to complete this task.
2. Make sure that the new deployment manager is running.
3. Identify the profiles involved.
   a. Identify an older version profile that contains cluster members.

   b. Identify which other clusters this profile contributes to; that is, if the profile defines servers that are members of any other clusters, identify those clusters.
   c. Identify all other profiles in the same cell that contribute cluster members to any of the clusters identified in step 3b.
   d. Identify all the node agents and process servers defined by any of the profiles identified in step 3c.

   All of the profiles identified in step 3c, and all of the corresponding node agents and servers identified in step 3d will be involved in the migration.

4. Stop all the node agents and servers identified in step 3d.
5. Migrate each profile identified in step 3c, one at a time, but **do not start** any new agents or servers. To do so, follow one of the instruction sets listed in either "Migrating cluster members using the migration wizard" on page 96 or "Migrating cluster members using the command-line tools" on page 100.
6. On the system that hosts the WebSphere Process Server version 6.2 deployment manager profile, navigate to the *install_dir*/util directory. This directory contains the WBIProfileUpgrade script, WBIProfileUpgrade.ant.
7. Run WBIProfileUpgrade for each cluster defined in step 3b. For instructions on running WBIProfileUpgrade, see WBIProfileUpgrade script.
8. If you are migrating a cluster that has Business Process Choreographer or Business Space configured, you must upgrade the associated databases manually at this point. See "Upgrading the Business Process Choreographer database manually" on page 38 or "Upgrading the Business Space database manually" on page 47.
9. Start all the new (migrated) node agents and servers that are members of the cluster.

Results

The cluster is now migrated to the new version of WebSphere Process Server.

What to do next

Verify that the migration has been successful. If your cluster has Business Process Choreographer configured, see "Postmigration tasks for Business Process Choreographer" on page 123. If your server has Business Space configured, see "Postmigration tasks for Business Space powered by WebSphere" on page 127. Finally, perform the checks described in "Postmigration tasks for WebSphere Process Server" on page 122.

Related concepts

Premigration considerations for Business Process Choreographer
If your servers run Business Process Choreographer, you should be aware of certain things that you need to plan and take into consideration before you migrate Business Process Choreographer.

Related tasks

Migrating a deployment manager
Migrate a WebSphere Process Server deployment manager, choosing from several methods depending on your needs.

Migrating non-clustered managed nodes
Migrate a WebSphere Process Server managed node, choosing from several methods depending on your needs.

Migrating a cluster with minimal down time
To migrate a cluster while minimizing down time, first migrate approximately half of the profiles contributing to the cluster, then migrate the other half. Perform the extra steps required for cluster migration after you have migrated the first set of profiles.

Upgrading the Business Process Choreographer database manually
After migrating a server or cluster that has Business Process Choreographer configured, you must manually upgrade the schema for the Business Process Choreographer database and perform a data migration before you start the server or any cluster member.

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

Postmigration tasks for WebSphere Process Server
After migration, you should check some configuration settings. You might need to change them, or further configure the version 6.2 server.

Related reference

WBIProfileUpgrade script
Use the WBIProfileUpgrade script to update application and configuration settings in a WebSphere Process Server profile when you are migrating clusters and in some other special situations.

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

Migrating cluster members using the migration wizard:

Migrate cluster members from an older version to a newer version of WebSphere Process Server using the migration wizard.

Before you begin

Note: The migration wizard cannot run in a non-graphical environment. Examples of non-graphical environments include the **i5/OS platform** or telnet sessions. If you want to run migration in a non-graphical environment, use the WBIPreUpgrade and WBIPostUpgrade commands.

Note: The migration wizard supports only WebSphere Process Server profiles. If you have WebSphere Application Server profiles, you must use the migration commands.

Note: These instructions are part of the larger procedure to migrate all the servers in your cluster. Follow the instructions in "Migrating a cluster" on page 94 or "Migrating a cluster with minimal down time" on page 104 before performing the steps described here.

Make sure that the following conditions are met before you start the migration process:

- Your system meets all the hardware and software requirements for the new version of WebSphere Process Server.
- If you are migrating on the same physical computer system on which the older version of WebSphere Process Server resides, you have installed the new version of WebSphere Process Server side-by-side the previous version on the same system.
- A federated profile, created with the older WebSphere Process Server version resides on the same system.
- Sufficient disk space is available for the migrated profile and its backup. See "Premigration considerations for WebSphere Process Server" on page 4 for disk space requirements.
- The deployment manager that manages the managed node that you intend to migrate has already been migrated to the newer version of WebSphere Process Server, and is running.

  Note: Migrating a WebSphere Process Server version 6.1.x or 6.0.2.x managed node to a version 6.2 managed node requires that you first migrate the version 6.1.x or 6.0.2.x deployment manager to a version 6.2 deployment manager. See "Migrating a deployment manager" on page 70 for instructions. Complete the migration of the deployment manager before you proceed with the instructions in this topic.

Make sure that you have completed the following tasks before you start the migration process:

- Back up the databases that support version 6.1.x or 6.0.2.x WebSphere Process Server components.

About this task

After you migrate a version 6.0.2.x deployment manager to a newer version of WebSphere Process Server, the newer version deployment manager runs in compatibility mode by default, where it can manage both older and newer versions of WebSphere Process Server. For example, after migration, a version 6.2 deployment manager can manage both version 6.0.2.x and version 6.2 nodes. In other words, version 6.0.2.x managed nodes can run with the version 6.2 deployment manager. Over time, you can migrate each version 6.0.2.x WebSphere Process Server managed node (server managed by a version 6.2 deployment manager) to a version 6.2 managed node. After migrating all version 6.0.2.x managed nodes, you use the `convertScriptCompatibility` script to convert their configurations from a mode that supports compatibility of version 6.0.2.x administration scripts to a mode that is fully in a configuration version 6.2 model. See the convertScriptCompatibility command.

Note: When following the directions at this link to use the convertScriptCompatibility command, use the WBIPostUpgrade command rather than the WASPostUpgrade command.

Procedure

1. Log on as the root user on a Linux or UNIX system, or as a member of the Administrator group on a Windows system.
2. Stop the version 6.1.x or version 6.0.2.x server if it is running on the node to be migrated. Use the stopServer command from the *profile_dir*/bin directory for the profile of the affected server, or stop the server from the profile's First steps console.

   For more information about the stopServer command see the stopServer command. Use the following syntax:

   - Linux UNIX **On Linux and UNIX platforms:** *profile_root*/bin/stopServer.sh *server_name*
   - Windows **On Windows platforms:** *profile_root*\bin\stopServer.bat *server_name*

   If security is enabled, use one of the following commands instead. The user name provided must be a member of the operator or administrator role.

   - Linux UNIX **On Linux and UNIX platforms:** *profile_root*/bin/stopServer.sh *server_name* -username *user_ID* -password *password*
   - Windows **On Windows platforms:** *profile_root*\bin\stopServer.bat *server_name* -username *user_ID* -password *password*

   On the Windows operating system, even if security is enabled, the -username and -password parameters do not have to be specified if the server is running as a Windows service. In this case, the parameters are automatically passed into the script that the Windows service uses to shut down the system.

   Note: Before you start the migration process, you must stop the server from which you are migrating. It is not necessary to have that server running in order to migrate its configuration. The migration tools can retrieve all of the configuration data while the server is stopped.
3. Stop the node agent of the node to be migrated. Issue one of the following commands to stop the nodeagent process, depending on platform (where *profile_root* represents the installation directory of the federated node):

   - Linux UNIX **On Linux and UNIX platforms:** *profile_root*/bin/stopNode.sh
   - Windows **On Windows platforms:** *profile_root*\bin\stopNode.bat

   If security is enabled, use one of the following commands instead:

   - Linux UNIX **On Linux and UNIX platforms:** *profile_root*/bin/stopNode.sh -username *user_ID* -password *password*
   - Windows **On Windows platforms:** *profile_root*\bin\stopNode.bat -username *user_ID* -password *password*
4. Identify, in advance, the pre-existing information required for the migration, as listed below:

   Installation root directory
   See WBIPreUpgrade command-line utility for a description of the currentWebSphereDirectory parameter.

   Migration backup directory name
   See WBIPreUpgrade command-line utility for a description of the backupDirectory parameter.

Administrative security user name (required if adminstrative security is configured)
See WBIPostUpgrade command-line utility for a description of the -username parameter.

Administrative security password (required if adminstrative security is configured)
See WBIPostUpgrade command-line utility for a description of the -password parameter.

Source profile name
See WBIPostUpgrade command-line utility for a description of the -oldProfile parameter.

Target profile name
See WBIPostUpgrade command-line utility for a description of the -profileName parameter.

Port value assignments (optional)
See WBIPostUpgrade command-line utility for a description of the -replacePorts and -portBlock parameters.

Note: This applies only if you are migrating from version 6.0.2.x toversion 6.2.

5. Ensure that the version 6.2 deployment manager is up and running.
6. Invoke the migration wizard.

   Invoke the migration wizard in one of the following ways:

   - From the WebSphere Process Server First Steps console, select **Migration wizard**.
   - Run one of the following scripts (depending upon your operating system) stored in the *install_dir*/bin directory:
     - Linux UNIX **On Linux and UNIX platforms:** `wbi_migration.sh`
     - Windows **On Windows platforms:** `wbi_migration.bat`

     Note: You can optionally change the default trace setting (`*=all=enabled:com.ibm.ws.migration.common.*=all=disabled`) when invoking the migration wizard. The default trace setting enables tracing on only certain classes, but you can change the default to either enable full tracing or disable all tracing.

     - To enable full tracing, run one of the following scripts to invoke the migration wizard, depending on your operating system:
       - Linux UNIX **On Linux and UNIX platforms:** `wbi_migration.sh -W -migrationPanel.traceString="*=all=enabled"`
       - Windows **On Windows platforms:** `wbi_migration.bat -W -migrationPanel.traceString="*=all=enabled"`
     - To disable all tracing, run one of the following scripts to invoke the migration wizard, depending on your operating system:
       - Linux UNIX **On Linux and UNIX platforms:** `wbi_migration.sh -W -migrationPanel.traceString="*=all=disabled"`
       - Windows **On Windows platforms:** `wbi_migration.bat -W -migrationPanel.traceString="*=all=disabled"`

The migration wizard copies the configuration and applications from the version 6.1.x or 6.0.2.x managed node to the version 6.2 managed node. After migrating all of the data, the wizard federates the version 6.2 managed node into the deployment manager cell.

7. Repeat steps 1-6 on page 84 for each cluster member you wish to migrate.
8. If you chose the compatibility option (which is the default), and if all of your nodes are completely migrated to WebSphere Process Server version 6.2, run the convertScriptCompatibility script to remove compatibility from the version 6.2 deployment manager.

   Note: This applies only if you are migrating from version 6.0.2.x.

   Issue the convertScriptCompatibility command from the bin directory.

   - UNIX Linux *install_root*/bin/convertScriptCompatibility.sh
   - Windows *install_root*\bin\convertScriptCompatibility.bat

   See the convertScriptCompatibility command.

Results

The cluster member profiles are now migrated.

What to do next

Complete the cluster migration by completing steps 6-9 in "Migrating a cluster" on page 94 or steps 7-12 in "Migrating a cluster with minimal down time" on page 104.

Related tasks

Migrating cluster members using the command-line tools
Migrate cluster members from an older version to a newer version of WebSphere Process Server with the command-line tools.

Migrating cluster members using the command-line tools:

Migrate cluster members from an older version to a newer version of WebSphere Process Server with the command-line tools.

Before you begin

Note: These instructions are part of the larger procedure to migrate all the servers in your cluster. Follow the instructions in "Migrating a cluster" on page 94 or "Migrating a cluster with minimal down time" on page 104 before performing the steps described here.

Note: When you migrate using the command-line tools, you can migrate either a WebSphere Process Server profile or a WebSphere Application Server profile.

Make sure that the following conditions are met before you start the migration process:

- Your system meets all the hardware and software requirements for the new version of WebSphere Process Server.
- If you are migrating on the same physical computer system on which the older version of WebSphere Process Server resides, you have installed the new version of WebSphere Process Server side-by-side the previous version on the same system.

- A federated profile, created with the older WebSphere Process Server version resides on the same system.
- Sufficient disk space is available for the migrated profile and its backup. See "Premigration considerations for WebSphere Process Server" on page 4 for disk space requirements.
- The deployment manager that manages the managed node that you intend to migrate has already been migrated to the newer version of WebSphere Process Server, and is running.

  Note: Migrating a WebSphere Process Server version 6.1.x or 6.0.2.x managed node to a version 6.2 managed node requires that you first migrate the version 6.1.x or 6.0.2.x deployment manager to a version 6.2 deployment manager. See "Migrating a deployment manager" on page 70 for instructions. Complete the migration of the deployment manager before you proceed with the instructions in this topic.

Make sure that you have completed the following tasks before you start the migration process:
- Back up the databases that support version 6.1.x or 6.0.2.x WebSphere Process Server components.

About this task

After you migrate a version 6.0.2.x deployment manager to a newer version of WebSphere Process Server, the newer version deployment manager runs in compatibility mode by default, where it can manage both older and newer versions of WebSphere Process Server. For example, after migration, a version 6.2 deployment manager can manage both version 6.0.2.x and version 6.2 nodes. In other words, version 6.0.2.x managed nodes can run with the version 6.2 deployment manager. Over time, you can migrate each version 6.0.2.x WebSphere Process Server managed node (server managed by a version 6.2 deployment manager) to a version 6.2 managed node. After migrating all version 6.0.2.x managed nodes, you use the `convertScriptCompatibility` script to convert their configurations from a mode that supports compatibility of version 6.0.2.x administration scripts to a mode that is fully in a configuration version 6.2 model. See the convertScriptCompatibility command.

Note: When following the directions at this link to use the convertScriptCompatibility command, use the WBIPostUpgrade command rather than the WASPostUpgrade command.

Procedure

1. Log on as the root user on a Linux or UNIX system, or as a member of the Administrator group on a Windows system.
2. Stop the version 6.1.x or 6.0.2.x server if it is running on the node to be migrated. Use the stopServer command from the *profile_dir*/bin directory for the profile of the affected server, or stop the server from the profile's First steps console.

   For more information about the stopServer command see the stopServer command. Use the following syntax:

   Note: On i5/OS platforms, you must run the scripts under QSHELL. To start a QSHELL session, open a CL command prompt and type `QSH`.

   - i5/OS **On i5/OS platforms:** `*profile_root*/bin/stopServer *server_name*`

- Linux UNIX **On Linux and UNIX platforms:** *profile_root*/bin/stopServer.sh *server_name*
- Windows **On Windows platforms:** *profile_root*\bin\stopServer.bat *server_name*

If security is enabled, use one of the following commands instead. The user name provided must be a member of the operator or administrator role.

- i5/OS **On i5/OS platforms:** *profile_root*/bin/stopServer *server_name* -username *user_ID* -password *password*
- Linux UNIX **On Linux and UNIX platforms:** *profile_root*/bin/stopServer.sh *server_name* -username *user_ID* -password *password*
- Windows **On Windows platforms:** *profile_root*\bin\stopServer.bat *server_name* -username *user_ID* -password *password*

On the Windows operating system, even if security is enabled, the -username and -password parameters do not have to be specified if the server is running as a Windows service. In this case, the parameters are automatically passed into the script that the Windows service uses to shut down the system.

Note: Stop the server before beginning the migrating process. By default, all servers on the node are stopped before the migration completes.

3. Stop the node agent of the node to be migrated. Issue one of the following commands to stop the nodeagent process, depending on platform (where *profile_root* represents the installation directory of the federated node):

- i5/OS **On i5/OS platforms:** *profile_root*/bin/stopNode
- Linux UNIX **On Linux and UNIX platforms:** *profile_root*/bin/stopNode.sh
- Windows **On Windows platforms:** *profile_root*\bin\stopNode.bat

If security is enabled, use one of the following commands instead:

- i5/OS **On i5/OS platforms:** *profile_root*/bin/stopNode -username *user_ID* -password *password*
- Linux UNIX **On Linux and UNIX platforms:** *profile_root*/bin/stopNode.sh -username *user_ID* -password *password*
- Windows **On Windows platforms:** *profile_root*\bin\stopNode.bat -username *user_ID* -password *password*

Note: You must stop the old node before you start migration process. It is not necessary to have the server running to migrate its configuration. The migration tools can retrieve all the configuration data while the server is stopped.

4. Identify, in advance, the pre-existing information required for the migration, as listed below:

Installation root directory
See WBIPreUpgrade command-line utility for a description of the currentWebSphereDirectory parameter.

Migration backup directory name
See WBIPreUpgrade command-line utility for a description of the backupDirectory parameter.

Administrative security user name (required if adminstrative security is configured)
See WBIPostUpgrade command-line utility for a description of the -username parameter.

Administrative security password (required if adminstrative security is configured)
See WBIPostUpgrade command-line utility for a description of the -password parameter.

Source profile name
See WBIPostUpgrade command-line utility for a description of the -oldProfile parameter.

Target profile name
See WBIPostUpgrade command-line utility for a description of the -profileName parameter.

Port value assignments (optional)
See WBIPostUpgrade command-line utility for a description of the -replacePorts and -portBlock parameters.

Note: This applies only if you are migrating from version 6.0.2.x toversion 6.2.

5. Ensure that the version 6.2 deployment manager is up and running.
6. Run the WBIPreUpgrade command, specifying the migration backup directory name and the existing WebSphere Process Server directory name. The WBIPreUpgrade tool saves the configuration files of your existing profiles to the backup directory you specify.
7. Run the WBIPostUpgrade command, specifying the migration backup directory. The WBIPostUpgrade tool restores the backed up configuration in the backup directory into the new WebSphere Process Server Deployment Manager profile.

   Important: Use the `-createTargetProfile` parameter when invoking WBIPostUpgrade. This option creates a matching required new target profile for migration. For more information about target profiles, refer to “Target profile considerations” on page 9.

   Note: i5/OS If you are migrating on an **i5/OS platform**, the target profile name must match the profile name of the source profile being migrated.
8. Repeat steps 1-7 (with the possible exception of step 6).

   Note: You need to perform step 6 (running WBIPreUpgrade) again only you are migrating from version 6.1.x, or if you are migrating from version 6.0.2.x and the version 6.0.2.x system has been reconfigured since the first time you ran WBIPreUpgrade. If you skip step 7 because you are migrating additional managed profiles in the same WebSphere Process Server installation, then you may also be able to skip step 1.
9. Linux UNIX Windows If you chose the compatibility option (which is the default), and if all of your nodes are completely migrated to WebSphere Process Server version 6.2, run the convertScriptCompatibility script to remove compatibility from the version 6.2 deployment manager.

   Note: Perform this step only if you are migrating from version 6.0.2.x.

   Note: This step does is not applicable to i5/OS platforms.

   Issue the convertScriptCompatibility command from the bin directory.

- UNIX Linux *install_root*/bin/convertScriptCompatibility.sh
- Windows *install_root*\bin\convertScriptCompatibility.bat

See the convertScriptCompatibility command.

Results

The cluster member profiles are now migrated.

What to do next

Complete the cluster migration by completing steps 6-9 in “Migrating a cluster” on page 94 or steps 7-12 in “Migrating a cluster with minimal down time.”

Related tasks

Migrating cluster members using the migration wizard
Migrate cluster members from an older version to a newer version of WebSphere Process Server using the migration wizard.

Migrating a cluster with minimal down time

To migrate a cluster while minimizing down time, first migrate approximately half of the profiles contributing to the cluster, then migrate the other half. Perform the extra steps required for cluster migration after you have migrated the first set of profiles.

Before you begin

You must have an existing cell containing at least one cluster running on an older version of WebSphere Process Server (for example, version 6.1.x or 6.0.2.x) that you wish to migrate to a newer version (for example, version 6.2). In addition, you must have installed the new version of WebSphere Process Server.

Important: In a cluster, version 6.1.x or 6.0.2.x members and version 6.2 members must never run at the same time. All version 6.1.x or 6.0.2.x cluster members must be stopped before you start the first version 6.2 cluster member. Also, once you start a version 6.2 cluster member, do not start any version 6.0.2.x cluster members in that cluster.

About this task

Following these steps will ensure that you retain cluster functionality on the new version of WebSphere Process Server with minimal downtime.

Restriction: The following procedure is supported only if you are migrating from version 6.1.x to version 6.2. If you are migrating from version 6.0.2.x, and you want to minimize downtime while migrating a cluster, you must migrate to version 6.1.x first, then migrate to version 6.2.

Procedure

1. Migrate the deployment manager. Follow one of the instruction sets listed in “Migrating a deployment manager” on page 70 to complete this task.
2. Make sure that the new deployment manager is running.
3. Identify the profiles involved.
   a. Identify an older version profile that contains cluster members.

b. Identify which other clusters this profile contributes to; that is, if the profile defines servers that are members of any other clusters, identify those clusters.

c. Identify all other profiles in the same cell that contribute cluster members to any of the clusters identified in step 3b.

d. Identify all the node agents and process servers defined by any of the profiles identified in step 3c.

All of the profiles identified in step 3c, and all of the corresponding node agents and servers identified in step 3d will be involved in the migration.

4. Define two groups of profiles out of the full set of profiles identified in step 3 on page 94. Divide the profiles into roughly half (If the total number of profiles is an odd number, one group will consist of one more in number than the other group). You will migrate one set of servers while the other set is still running, thus reducing the amount of time when all servers in the cluster are stopped.
5. Stop all the node agents and servers that are defined by the first set of profiles that you will migrate.
6. Migrate each profile in the first set, one at a time, but **do not start** any new node agents or servers. Follow one of the instruction sets listed in either “Migrating cluster members using the migration wizard” on page 96 or “Migrating cluster members using the command-line tools” on page 100.
7. Stop the remaining node agents and servers; that is, those defined by the second set of profiles. This action starts the period of time when cluster services will be unavailable.
8. On the system that hosts the WebSphere Process Server version 6.2 deployment manager profile, navigate to the *install_dir*/util directory. This directory contains the WBIProfileUpgrade script, WBIProfileUpgrade.ant.
9. Run the WBIProfileUpgrade script for each cluster defined in the profiles that have been migrated so far. (That is, run WBIProfileUpgrade for each cluster defined in step 3 on page 94.) For instructions on running WBIProfileUpgrade, see WBIProfileUpgrade script.
10. If you are migrating a cluster that has Business Process Choreographer or Business Space configured, you must upgrade the associated databases manually at this point. See “Upgrading the Business Process Choreographer database manually” on page 38 or “Upgrading the Business Space database manually” on page 47.
11. Start all the new (migrated) node agents and servers; that is, the node agents and servers corresponding to the profiles that have been migrated so far.
12. Migrate each profile in the second set of profiles As with the first set, to migrate, follow one of the instruction sets listed in either “Migrating cluster members using the migration wizard” on page 96 or “Migrating cluster members using the command-line tools” on page 100. This time, you can start the migrated node agents and servers as you proceed to migrate each managed node.

Results

The cluster is now migrated to the new version of WebSphere Process Server.

What to do next

Verify that the migration has been successful. If your cluster has Business Process Choreographer configured, see “Postmigration tasks for Business Process

Choreographer” on page 123. If your cluster has Business Space configured, see “Postmigration tasks for Business Space powered by WebSphere” on page 127. Finally, perform the checks described in “Postmigration tasks for WebSphere Process Server” on page 122.

Related concepts

Premigration considerations for Business Process Choreographer
If your servers run Business Process Choreographer, you should be aware of certain things that you need to plan and take into consideration before you migrate Business Process Choreographer.

Related tasks

Migrating a deployment manager
Migrate a WebSphere Process Server deployment manager, choosing from several methods depending on your needs.

Migrating non-clustered managed nodes
Migrate a WebSphere Process Server managed node, choosing from several methods depending on your needs.

Migrating a cluster
To migrate a cluster, migrate each profile containing a member of that cluster one at a time. The migration requires extra steps not required for a non-clustered environment.

Upgrading the Business Process Choreographer database manually
After migrating a server or cluster that has Business Process Choreographer configured, you must manually upgrade the schema for the Business Process Choreographer database and perform a data migration before you start the server or any cluster member.

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

Postmigration tasks for WebSphere Process Server
After migration, you should check some configuration settings. You might need to change them, or further configure the version 6.2 server.

Related reference

WBIProfileUpgrade script
Use the WBIProfileUpgrade script to update application and configuration settings in a WebSphere Process Server profile when you are migrating clusters and in some other special situations.

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

Migrating cluster members using the migration wizard:

Migrate cluster members from an older version to a newer version of WebSphere Process Server using the migration wizard.

Before you begin

Note: The migration wizard cannot run in a non-graphical environment. Examples of non-graphical environments include the **i5/OS platform** or telnet sessions. If you want to run migration in a non-graphical environment, use the WBIPreUpgrade and WBIPostUpgrade commands.

Note: The migration wizard supports only WebSphere Process Server profiles. If you have WebSphere Application Server profiles, you must use the migration commands.

Note: These instructions are part of the larger procedure to migrate all the servers in your cluster. Follow the instructions in “Migrating a cluster” on page 94 or “Migrating a cluster with minimal down time” on page 104 before performing the steps described here.

Make sure that the following conditions are met before you start the migration process:

- Your system meets all the hardware and software requirements for the new version of WebSphere Process Server.
- If you are migrating on the same physical computer system on which the older version of WebSphere Process Server resides, you have installed the new version of WebSphere Process Server side-by-side the previous version on the same system.
- A federated profile, created with the older WebSphere Process Server version resides on the same system.
- Sufficient disk space is available for the migrated profile and its backup. See “Premigration considerations for WebSphere Process Server” on page 4 for disk space requirements.
- The deployment manager that manages the managed node that you intend to migrate has already been migrated to the newer version of WebSphere Process Server, and is running.

  Note: Migrating a WebSphere Process Server version 6.1.x or 6.0.2.x managed node to a version 6.2 managed node requires that you first migrate the version 6.1.x or 6.0.2.x deployment manager to a version 6.2 deployment manager. See “Migrating a deployment manager” on page 70 for instructions. Complete the migration of the deployment manager before you proceed with the instructions in this topic.

Make sure that you have completed the following tasks before you start the migration process:

- Back up the databases that support version 6.1.x or 6.0.2.x WebSphere Process Server components.

About this task

After you migrate a version 6.0.2.x deployment manager to a newer version of WebSphere Process Server, the newer version deployment manager runs in compatibility mode by default, where it can manage both older and newer versions of WebSphere Process Server. For example, after migration, a version 6.2 deployment manager can manage both version 6.0.2.x and version 6.2 nodes. In other words, version 6.0.2.x managed nodes can run with the version 6.2 deployment manager. Over time, you can migrate each version 6.0.2.x WebSphere Process Server managed node (server managed by a version 6.2 deployment manager) to a version 6.2 managed node. After migrating all version 6.0.2.x managed nodes, you use the `convertScriptCompatibility` script to convert their configurations from a mode that supports compatibility of version 6.0.2.x administration scripts to a mode that is fully in a configuration version 6.2 model. See the convertScriptCompatibility command.

Note: When following the directions at this link to use the convertScriptCompatibility command, use the WBIPostUpgrade command rather than the WASPostUpgrade command.

Procedure

1. Log on as the root user on a Linux or UNIX system, or as a member of the Administrator group on a Windows system.
2. Stop the version 6.1.x or version 6.0.2.x server if it is running on the node to be migrated. Use the stopServer command from the *profile_dir*/bin directory for the profile of the affected server, or stop the server from the profile's First steps console.

   For more information about the stopServer command see the stopServer command. Use the following syntax:

   - Linux UNIX **On Linux and UNIX platforms:** *profile_root*/bin/stopServer.sh *server_name*
   - Windows **On Windows platforms:** *profile_root*\bin\stopServer.bat *server_name*

   If security is enabled, use one of the following commands instead. The user name provided must be a member of the operator or administrator role.

   - Linux UNIX **On Linux and UNIX platforms:** *profile_root*/bin/stopServer.sh *server_name* -username *user_ID* -password *password*
   - Windows **On Windows platforms:** *profile_root*\bin\stopServer.bat *server_name* -username *user_ID* -password *password*

   On the Windows operating system, even if security is enabled, the -username and -password parameters do not have to be specified if the server is running as a Windows service. In this case, the parameters are automatically passed into the script that the Windows service uses to shut down the system.

   Note: Before you start the migration process, you must stop the server from which you are migrating. It is not necessary to have that server running in order to migrate its configuration. The migration tools can retrieve all of the configuration data while the server is stopped.
3. Stop the node agent of the node to be migrated. Issue one of the following commands to stop the nodeagent process, depending on platform (where *profile_root* represents the installation directory of the federated node):

   - Linux UNIX **On Linux and UNIX platforms:** *profile_root*/bin/stopNode.sh
   - Windows **On Windows platforms:** *profile_root*\bin\stopNode.bat

   If security is enabled, use one of the following commands instead:

   - Linux UNIX **On Linux and UNIX platforms:** *profile_root*/bin/stopNode.sh -username *user_ID* -password *password*
   - Windows **On Windows platforms:** *profile_root*\bin\stopNode.bat -username *user_ID* -password *password*
4. Identify, in advance, the pre-existing information required for the migration, as listed below:

   Installation root directory

   See WBIPreUpgrade command-line utility for a description of the currentWebSphereDirectory parameter.

Migration backup directory name
: See WBIPreUpgrade command-line utility for a description of the backupDirectory parameter.

Administrative security user name (required if adminstrative security is configured)
: See WBIPostUpgrade command-line utility for a description of the -username parameter.

Administrative security password (required if adminstrative security is configured)
: See WBIPostUpgrade command-line utility for a description of the -password parameter.

Source profile name
: See WBIPostUpgrade command-line utility for a description of the -oldProfile parameter.

Target profile name
: See WBIPostUpgrade command-line utility for a description of the -profileName parameter.

Port value assignments (optional)
: See WBIPostUpgrade command-line utility for a description of the -replacePorts and -portBlock parameters.

Note: This applies only if you are migrating from version 6.0.2.x toversion 6.2.

5. Ensure that the version 6.2 deployment manager is up and running.
6. Invoke the migration wizard.

   Invoke the migration wizard in one of the following ways:
   - From the WebSphere Process Server First Steps console, select **Migration wizard**.
   - Run one of the following scripts (depending upon your operating system) stored in the *install_dir*/bin directory:
     – Linux UNIX **On Linux and UNIX platforms:** `wbi_migration.sh`
     – Windows **On Windows platforms:** `wbi_migration.bat`

     Note: You can optionally change the default trace setting (`*=all=enabled:com.ibm.ws.migration.common.*=all=disabled`) when invoking the migration wizard. The default trace setting enables tracing on only certain classes, but you can change the default to either enable full tracing or disable all tracing.
     – To enable full tracing, run one of the following scripts to invoke the migration wizard, depending on your operating system:
       - Linux UNIX **On Linux and UNIX platforms:** `wbi_migration.sh -W -migrationPanel.traceString="*=all=enabled"`
       - Windows **On Windows platforms:** `wbi_migration.bat -W -migrationPanel.traceString="*=all=enabled"`

     – To disable all tracing, run one of the following scripts to invoke the migration wizard, depending on your operating system:
       - Linux UNIX **On Linux and UNIX platforms:** `wbi_migration.sh -W -migrationPanel.traceString="*=all=disabled"`
       - Windows **On Windows platforms:** `wbi_migration.bat -W -migrationPanel.traceString="*=all=disabled"`

The migration wizard copies the configuration and applications from the version 6.1.x or 6.0.2.x managed node to the version 6.2 managed node. After migrating all of the data, the wizard federates the version 6.2 managed node into the deployment manager cell.

7. Repeat steps 1-6 on page 84 for each cluster member you wish to migrate.
8. If you chose the compatibility option (which is the default), and if all of your nodes are completely migrated to WebSphere Process Server version 6.2, run the convertScriptCompatibility script to remove compatibility from the version 6.2 deployment manager.

   Note: This applies only if you are migrating from version 6.0.2.x.

   Issue the convertScriptCompatibility command from the bin directory.
   - UNIX Linux *install_root*/bin/convertScriptCompatibility.sh
   - Windows *install_root*\bin\convertScriptCompatibility.bat

   See the convertScriptCompatibility command.

Results

The cluster member profiles are now migrated.

What to do next

Complete the cluster migration by completing steps 6-9 in “Migrating a cluster” on page 94 or steps 7-12 in “Migrating a cluster with minimal down time” on page 104.

Related tasks

Migrating cluster members using the command-line tools
Migrate cluster members from an older version to a newer version of WebSphere Process Server with the command-line tools.

Migrating cluster members using the command-line tools:

Migrate cluster members from an older version to a newer version of WebSphere Process Server with the command-line tools.

Before you begin

Note: These instructions are part of the larger procedure to migrate all the servers in your cluster. Follow the instructions in “Migrating a cluster” on page 94 or “Migrating a cluster with minimal down time” on page 104 before performing the steps described here.

Note: When you migrate using the command-line tools, you can migrate either a WebSphere Process Server profile or a WebSphere Application Server profile.

Make sure that the following conditions are met before you start the migration process:

- Your system meets all the hardware and software requirements for the new version of WebSphere Process Server.
- If you are migrating on the same physical computer system on which the older version of WebSphere Process Server resides, you have installed the new version of WebSphere Process Server side-by-side the previous version on the same system.

- A federated profile, created with the older WebSphere Process Server version resides on the same system.
- Sufficient disk space is available for the migrated profile and its backup. See "Premigration considerations for WebSphere Process Server" on page 4 for disk space requirements.
- The deployment manager that manages the managed node that you intend to migrate has already been migrated to the newer version of WebSphere Process Server, and is running.

  Note: Migrating a WebSphere Process Server version 6.1.x or 6.0.2.x managed node to a version 6.2 managed node requires that you first migrate the version 6.1.x or 6.0.2.x deployment manager to a version 6.2 deployment manager. See "Migrating a deployment manager" on page 70 for instructions. Complete the migration of the deployment manager before you proceed with the instructions in this topic.

Make sure that you have completed the following tasks before you start the migration process:

- Back up the databases that support version 6.1.x or 6.0.2.x WebSphere Process Server components.

About this task

After you migrate a version 6.0.2.x deployment manager to a newer version of WebSphere Process Server, the newer version deployment manager runs in compatibility mode by default, where it can manage both older and newer versions of WebSphere Process Server. For example, after migration, a version 6.2 deployment manager can manage both version 6.0.2.x and version 6.2 nodes. In other words, version 6.0.2.x managed nodes can run with the version 6.2 deployment manager. Over time, you can migrate each version 6.0.2.x WebSphere Process Server managed node (server managed by a version 6.2 deployment manager) to a version 6.2 managed node. After migrating all version 6.0.2.x managed nodes, you use the `convertScriptCompatibility` script to convert their configurations from a mode that supports compatibility of version 6.0.2.x administration scripts to a mode that is fully in a configuration version 6.2 model. See the convertScriptCompatibility command.

Note: When following the directions at this link to use the convertScriptCompatibility command, use the WBIPostUpgrade command rather than the WASPostUpgrade command.

Procedure

1. Log on as the root user on a Linux or UNIX system, or as a member of the Administrator group on a Windows system.
2. Stop the version 6.1.x or 6.0.2.x server if it is running on the node to be migrated. Use the stopServer command from the *profile_dir*/bin directory for the profile of the affected server, or stop the server from the profile's First steps console.

   For more information about the stopServer command see the stopServer command. Use the following syntax:

   Note: On i5/OS platforms, you must run the scripts under QSHELL. To start a QSHELL session, open a CL command prompt and type `QSH`.

   - i5/OS **On i5/OS platforms:** *profile_root*`/bin/stopServer` *server_name*

- Linux UNIX **On Linux and UNIX platforms:** *profile_root*/bin/stopServer.sh *server_name*
- Windows **On Windows platforms:** *profile_root*\bin\stopServer.bat *server_name*

If security is enabled, use one of the following commands instead. The user name provided must be a member of the operator or administrator role.

- i5/OS **On i5/OS platforms:** *profile_root*/bin/stopServer *server_name* -username *user_ID* -password *password*
- Linux UNIX **On Linux and UNIX platforms:** *profile_root*/bin/stopServer.sh *server_name* -username *user_ID* -password *password*
- Windows **On Windows platforms:** *profile_root*\bin\stopServer.bat *server_name* -username *user_ID* -password *password*

On the Windows operating system, even if security is enabled, the -username and -password parameters do not have to be specified if the server is running as a Windows service. In this case, the parameters are automatically passed into the script that the Windows service uses to shut down the system.

Note: Stop the server before beginning the migrating process. By default, all servers on the node are stopped before the migration completes.

3. Stop the node agent of the node to be migrated. Issue one of the following commands to stop the nodeagent process, depending on platform (where *profile_root* represents the installation directory of the federated node):

- i5/OS **On i5/OS platforms:** *profile_root*/bin/stopNode
- Linux UNIX **On Linux and UNIX platforms:** *profile_root*/bin/stopNode.sh
- Windows **On Windows platforms:** *profile_root*\bin\stopNode.bat

If security is enabled, use one of the following commands instead:

- i5/OS **On i5/OS platforms:** *profile_root*/bin/stopNode -username *user_ID* -password *password*
- Linux UNIX **On Linux and UNIX platforms:** *profile_root*/bin/stopNode.sh -username *user_ID* -password *password*
- Windows **On Windows platforms:** *profile_root*\bin\stopNode.bat -username *user_ID* -password *password*

Note: You must stop the old node before you start migration process. It is not necessary to have the server running to migrate its configuration. The migration tools can retrieve all the configuration data while the server is stopped.

4. Identify, in advance, the pre-existing information required for the migration, as listed below:

Installation root directory

See WBIPreUpgrade command-line utility for a description of the currentWebSphereDirectory parameter.

Migration backup directory name

See WBIPreUpgrade command-line utility for a description of the backupDirectory parameter.

Administrative security user name (required if adminstrative security is configured)
See WBIPostUpgrade command-line utility for a description of the -username parameter.

Administrative security password (required if adminstrative security is configured)
See WBIPostUpgrade command-line utility for a description of the -password parameter.

Source profile name
See WBIPostUpgrade command-line utility for a description of the -oldProfile parameter.

Target profile name
See WBIPostUpgrade command-line utility for a description of the -profileName parameter.

Port value assignments (optional)
See WBIPostUpgrade command-line utility for a description of the -replacePorts and -portBlock parameters.

Note: This applies only if you are migrating from version 6.0.2.x toversion 6.2.

5. Ensure that the version 6.2 deployment manager is up and running.
6. Run the WBIPreUpgrade command, specifying the migration backup directory name and the existing WebSphere Process Server directory name. The WBIPreUpgrade tool saves the configuration files of your existing profiles to the backup directory you specify.
7. Run the WBIPostUpgrade command, specifying the migration backup directory. The WBIPostUpgrade tool restores the backed up configuration in the backup directory into the new WebSphere Process Server Deployment Manager profile.

   Important: Use the `-createTargetProfile` parameter when invoking WBIPostUpgrade. This option creates a matching required new target profile for migration. For more information about target profiles, refer to "Target profile considerations" on page 9.

   Note: i5/OS If you are migrating on an **i5/OS platform**, the target profile name must match the profile name of the source profile being migrated.
8. Repeat steps 1-7 (with the possible exception of step 6).

   Note: You need to perform step 6 (running WBIPreUpgrade) again only you are migrating from version 6.1.x, or if you are migrating from version 6.0.2.x and the version 6.0.2.x system has been reconfigured since the first time you ran WBIPreUpgrade. If you skip step 7 because you are migrating additional managed profiles in the same WebSphere Process Server installation, then you may also be able to skip step 1.
9. Linux UNIX Windows If you chose the compatibility option (which is the default), and if all of your nodes are completely migrated to WebSphere Process Server version 6.2, run the convertScriptCompatibility script to remove compatibility from the version 6.2 deployment manager.

   Note: Perform this step only if you are migrating from version 6.0.2.x.

   Note: This step does is not applicable to i5/OS platforms.

   Issue the convertScriptCompatibility command from the bin directory.

- UNIX Linux *install_root*/bin/convertScriptCompatibility.sh
- Windows *install_root*\bin\convertScriptCompatibility.bat

See the convertScriptCompatibility command.

Results

The cluster member profiles are now migrated.

What to do next

Complete the cluster migration by completing steps 6-9 in “Migrating a cluster” on page 94 or steps 7-12 in “Migrating a cluster with minimal down time” on page 104.

Related tasks

Migrating cluster members using the migration wizard
Migrate cluster members from an older version to a newer version of WebSphere Process Server using the migration wizard.

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 6.2. 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 6.2 and the Business Rules Manager running on that deployment target has not yet been upgraded to version 6.2. 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 6.2 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 6.2 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 6.2, 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	`install_root/bin/wsadmin -f installBRManager.jacl -s server_name -n node_name`
a cluster	`install_root/bin/wsadmin -f installBRManager.jacl -cl cluster_name`

multiple targets (if the current Business Rules Manager is mapped to more than one target)	`install_root/bin/wsadmin -f installBRManager.jacl -m "{{target1} {target2} ... {targetn}}"`
	where each {*targetn*} is either { `-s` *server_name* `-n` *node_name*} or { `-cl` *cluster_name*}

What to do next

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 69.

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/WBIPostUpgrade.*profileName.timestamp*.log for either of the following messages:
      - `MIGR0259I: The migration has successfully completed.`
      - `MIGR0271W: Migration completed successfully, with one or more warnings.`

Note: *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.

Note: *profileName* is the name of the new profile you created in version 6.2 of WebSphere Process Server.

b. Check the file *backupDirectory*/logs/ WBIProfileUpgrade.ant.*profile_name*.*timestamp*.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 profile's log files for fatal profile creation or augmentation errors. Profile log files are located in the following directory: *install_root*/logs/ manageprofiles/*profile_name*.
3. Check the server log files.
   a. Navigate to the *profile_root*/logs/*server_name* directory corresponding to the migrated profile.
   b. Review the SystemOut.log file and make sure there are no fatal errors.
   c. Review the SystemErr.log file and make sure there are no fatal errors.
4. 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**.

      Note: **Test connection** does not work for ME datasources. To verify the connection for ME datasources, make sure there are no errors in the logs after the servers are started.
   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 140 for troubleshooting information.

Related tasks

Rolling back your environment
After migrating to a WebSphere Process Server version 6.2 environment, you can roll back to a version 6.1.x or 6.0.2.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

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

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.

Rolling back your environment

After migrating to a WebSphere Process Server version 6.2 environment, you can roll back to a version 6.1.x or 6.0.2.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 for WebSphere Process Server
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
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 6.2 deployment cell to version 6.1.x or 6.0.2.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 6.1.x or 6.0.2.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. (Optional) 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 6.1.x or 6.0.2.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 Network Deployment, version 6.1 information center.
- Run the **backupConfig** command or your own preferred utility to back up the version 6.1.x or 6.0.2.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 Network Deployment, version 6.1 information center.

3. Migrate the deployment cell.

Procedure

1. Stop all of the servers that are currently running in the WebSphere Process Server version 6.2 environment.
2. If you chose to disable the previous deployment manager when you migrated to the version 6.2 deployment manager, do one of the following:
   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 6.1.x or 6.0.2.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 Network Deployment, version 6.1 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 6.1.x or 6.0.2.x *profile_root*/bin directory of the deployment manager that you need to roll back from version 6.2.

      Linux In a Linux environment, for example, use the following parameters:

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

      Tip: If you have trouble running the migrationDisablementReversal.jacl script, try to manually go through 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 deployment manager node that you want to roll back.

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

         a) Delete or rename the serverindex.xml file.

         b) Rename the serverindex.xml_disabled file to serverindex.xml.
3. For each of the deployment cell's managed nodes that you need to roll back, do one of the following:

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 6.1.x or 6.0.2.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 Network Deployment, version 6.1 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 6.1.x or 6.0.2.x *profile_root*/bin directory of the managed node.

   Linux In a Linux environment, for example, use the following parameters:

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

   Tip: If you have trouble running the migrationDisablementReversal.jacl script, try to manually go through 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, do the following:

      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 6.2 deployment manager was running.

   See Synchronizing nodes with the wsadmin tool in the WebSphere Application Server Network Deployment, version 6.1 information center.

5. If you chose to keep the installed applications in the same location as the prior release during migration to version 6.2 and any of the version 6.2 applications are not compatible with the prior release, install applications that are compatible.

6. Delete the version 6.2 profiles.

   See Deleting a profile in the WebSphere Application Server Network Deployment, version 6.1 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 6.1.x or 6.0.2.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 6.2 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 using the migration wizard
Migrate a deployment manager from an older version to a newer version of WebSphere Process Server using the migration wizard.

Migrating a deployment manager using command-line tools
Migrate a deployment manager from an older version to a newer version of WebSphere Process Server using the command-line tools.

Related information

restoreConfig command

backupConfig command

Synchronizing nodes with the wsadmin tool

Deleting a profile

Rolling back a managed node

You can use the **restoreConfig** and **wsadmin** commands to roll back a migrated WebSphere Process Server version 6.2 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 6.1.x or 6.0.2.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 6.1.x or 6.0.2.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 Network Deployment, version 6.1 information center.
   - Run the **backupConfig** command or your own preferred utility to back up the version 6.1.x or 6.0.2.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 Network Deployment, version 6.1 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 6.2 deployment manager configuration as it was before you migrated the version 6.1.x or 6.0.2.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 117.

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 6.2 environment.
3. Restore your previous configuration.
   a. Run the **restoreConfig** command or your own preferred utility to restore the version 6.2 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 Network Deployment, version 6.1 Information Center.
   b. Perform one of the following actions to restore the version 6.1.x or 6.0.2.x configuration for the managed node.
      - Run the **restoreConfig** command or your own preferred utility to restore the version 6.1.x or 6.0.2.x configuration.

        See restoreConfig command in the WebSphere Application Server Network Deployment, version 6.1 Information Center.
      - Use the **wsadmin** command to run the migrationDisablementReversal.jacl script from the version 6.1.x or 6.0.2.x*profile_root*/bin directory of the managed node.

        Linux In a Linux environment, for example, 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:

           ```
           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.

4. Start the version 6.2 deployment manager.
5. Synchronize the managed node.

   See Synchronizing nodes with the wsadmin tool in the WebSphere Application Server Network Deployment, version 6.1 information center.
6. If you chose to keep the installed applications in the same location as the prior release during migration to version 6.2 and any of the version 6.2 applications are not compatible with the prior release, install applications that are compatible.
7. Delete the version 6.2 managed profile.

   See Deleting a profile in the WebSphere Application Server Network Deployment, version 6.1 Information Center.
8. Start the rolled-back managed node in the version 6.2 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
You can use the **restoreConfig** and **wsadmin** commands to roll back a migrated WebSphere Process Server version 6.2 deployment cell to version 6.1.x or 6.0.2.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 non-clustered managed nodes using the migration wizard
Migrate non-clustered managed nodes from an older version to a newer version of WebSphere Process Server using the migration wizard.

Migrating non-clustered managed nodes using the command-line tools
Migrate non-clustered managed nodes from an older version to a newer version of WebSphere Process Server with the command-line tools.

Related information

restoreConfig command

backupConfig command

Synchronizing nodes with the wsadmin tool

Deleting a profile

Postmigration tasks for WebSphere Process Server

After migration, you should check some configuration settings. You might need to change them, or further configure the version 6.2 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 6.1.x or 6.0.2.x, and make sure that version 6.2 security is set appropriately.
- Check the WBIPostUpgrade.*profile_name.timestamp*.log file in the logs directory for details about any JSP objects that the migration tools did not migrate.

  If version 6.2 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.
- Verify the results of the automatic Cloudscape database migration, and manually migrate any Cloudscape databases that are not automatically migrated by the tools. See "Migrating Cloudscape databases" on page 128 for more information.
- After migrating from version 6.1.x to version 6.2, you should check your WebSphere Adapter properties to ensure that they are properly configured for the new installation location. Some adapter properties may need to be altered during migration in a way that would be unknown to an automated migration.
- After migrating to version 6.2, you need to check that your ports are mapped correctly to make sure that the Remote Artifact Loader can access the security port on the application cluster when the global security is turned on. To verify that your ports are configured correctly, use the following procedure.
  1. In the administrative console, navigate to **Environment** → **Virtual Hosts.**
  2. Select **default_host** → **Host Aliases.**
  3. Check if the application cluster security port is mapped to ″*″ which means ″all hosts.″ If it is not, change it to ″*″ by clicking **New**, then entering ″*″ in the Host Name field and the port number of the application cluster in the Port field.
  4. Save your changes by clicking **Apply** or **OK**, and then select **Save**.

Postmigration tasks for Business Process Choreographer

If your servers or clusters run Business Process Choreographer, you must perform some additional tasks before you start your servers or clusters.

Before you begin

You have successfully upgraded the Business Process Choreographer database schema and migrated the runtime data. You have also successfully migrated your servers and clusters.

About this task

You must perform these tasks, if they apply to your environment, before using your WebSphere Process Serverversion 6.2 in production.

Procedure

1. If you have applied any changes to the default XSL transformation files (EverybodyTransformation.xsl, LDAPTransformation.xsl, SystemTransformation.xsl, VMMTransformation.xsl, and UserRegistryTransformation.xsl) located in the *install_root*/ ProcessChoreographer/Staff directory then you must re-apply your changes to the WebSphere Process Serverversion 6.2 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 6.1.x or 6.0.2.x staff plug-in configuration (now known as the people directory configuration in WebSphere Process Serverversion 6.2).
2. Configure the REST API endpoints for the Business Flow Manager and Human Task Manager, and update all references. Depending on the version that you migrated from, perform one of the following:
   - If you migrated from version 6.1.2, the endpoints are created automatically in the WebSphere configuration repository, so you do not need the bpcEndpoints.xml file anymore. However, your customization is lost, and Business Space is either using one of the cluster members or the stand-alone server instead of the Web server. If the REST Web modules were mapped to a Web server before migration they are still mapped to the Web server but you must change the reference in Business Space to point to the Web server again by performing the following:
     a. To change the endpoint for the Business Flow Manager, click either **Servers** → **Application servers** → *server_name* or **Servers** → **Clusters** → *cluster_name*, then under **Business Integration**, expand **Business Process Choreographer**, and click **Business Flow Manager**, and under **Additional Properties** click **REST Service Endpoint**.
     b. To change the endpoint for the Human Task Manager, click either **Servers** → **Application servers** → *server_name* or **Servers** → **Clusters** → *cluster_name*, then under **Business Integration**, expand **Business Process Choreographer**, and click **Human Task Manager**, and under **Additional Properties** click **REST Service Endpoint**.
   - If you migrated from version 6.0.2 or version 6.1, the REST APIs were configured during the migration. You might want to map the Web modules to a Web server and change the context root for the REST API Web modules. If you make these changes, you must update the references to the REST APIs in Business Process Choreographer Explorer and Business Space too.

     To change the context root:

     a. In the administrative console click **Applications** → **Enterprise Applications** → **BPEContainer_*suffix*** → **Context Root for Web Modules**. Where *suffix* is either *node_name_server_name* or the *cluster_name* where Business Process Choreographer is configured.
     b. Make sure that the context root for the Web module BFMRESTAPI is correct and unique.
     c. In the administrative console click **Applications** → **Enterprise Applications** → **TaskContainer_*suffix*** → **Context Root for Web Modules**
     d. Make sure that the context root for the Web module HTMRESTAPI is correct and unique.

To change the endpoint references for Business Process Choreographer Explorer, click **Servers** → **Application servers** → *server_name* or **Servers** → **Clusters** → *cluster_name*, then under **Business Integration**, expand **Business Process Choreographer**, and click **Business Process Choreographer Explorer**, then in the list of configured Business Process Choreographer Explorer instances, click one to edit it, and change the values for **Business Flow Manager REST API URL** and **Human Task Manager REST API URL**. Repeat this as necessary for the other instances.

To change the endpoint references for Business Space:

a. To change the endpoint for the Business Flow Manager, click either **Servers** → **Application servers** → *server_name* or **Servers** → **Clusters** → *cluster_name*, then under **Business Integration**, expand **Business Process Choreographer**, and click **Business Flow Manager**, and under **Additional Properties** click **REST Service Endpoint**.

b. To change the endpoint for the Human Task Manager, click either **Servers** → **Application servers** → *server_name* or **Servers** → **Clusters** → *cluster_name*, then under **Business Integration**, expand **Business Process Choreographer**, and click **Human Task Manager**, and under **Additional Properties** click **REST Service Endpoint**.

3. In a network deployment environment, make sure that the latest version of the predefined human tasks are installed. If they are not, you must install them manually.

   a. To check whether the latest version of the predefined human tasks are installed, click **Applications** → **Enterprise Applications**, then look for the applications named `HTM_PredefinedTasks_V`*`nnn_scope`* and `HTM_PredefinedTasksMsg_V`*`nnn_scope`*. If the version numbers *nnn*, are identical to those located at *install_root*/installableApps/HTM_PredefinedTasks_V*nnn*.ear and *install_root*/installableApps/HTM_PredefinedTasksMsg_V*nnn*.ear , then you have the latest versions, and you can skip to step 4 on page 126. Otherwise, perform the following actions to install the applications.

   b. Stop the deployment manager.

   c. On the deployment manager, change to the directory where the bpeupgrade.jacl script is located, and run the script.

   On Linux and UNIX platforms:
   Change to the directory *`install_root`*`/ProcessChoreographer/config` and enter the command:

   ```
   ../../bin/wsadmin.sh -conntype NONE -profileName profileName
                        -f bpeupgrade.jacl -cluster clusterName
   ```

   On i5/OS platforms:
   Change to the directory *`install_root`*`/ProcessChoreographer/config` and enter the command:

   ```
   ../../bin/wsadmin -conntype NONE -profileName profileName
                     -f bpeupgrade.jacl -cluster clusterName
   ```

   On Windows platforms:
   Change to the directory *`install_root`*`\ProcessChoreographer\config` and enter the command:

   ```
   ..\..\bin/wsadmin -conntype NONE -profileName profileName
                     -f bpeupgrade.jacl -cluster clusterName
   ```

   Where *profileName* is the name of the deployment manager's profile and *clusterName* is the name of cluster where Business Process Choreographer is configured.

d. Start the deployment manager.

e. Synchronize the configuration changes with the nodes and restart the cluster members.

4. When there are no more instances of the old versions of the predefined human tasks running, remove them:

   a. Make sure that all instances have been deleted.

   b. In the administrative console, click **Applications** → **Enterprise Applications** select both of the following applications, then click **Uninstall**.

      - HTM_PredefinedTasks_V*nnn_scope*.ear
      - HTM_PredefinedTaskMsg_V*nnn_scope*.ear

      where *nnn* is the old version number, for example 612, and *scope* is either *nodeName_serverName* or *clusterName*, depending on whether the predefined tasks are installed on a single server or on a cluster.

5. Optional: To release the extra storage space used by the work item data migration, delete the table WI_ASSOC_OID_T from the database.

6. Optional: If you use DB2 for Linux, UNIX, Windows, or z/OS, to release the extra storage space used by the table space migration, delete the following old tables from the database:

   - PROCESS_TEMPLATE_B_O
   - ACTIVITY_TEMPLATE_B_O
   - SCOPED_VARIABLE_INSTANCE_B_O
   - CORRELATION_SET_INSTANCE_B_O
   - STAFF_QUERY_INSTANCE_O
   - TASK_TEMPLATE_O
   - TASK_INSTANCE_O

   Attention: Take care not to delete any of the new tables, which have similar names but have the suffix "_T".

7. Optional: Retune your database now or later. For example, for DB2 databases, run REORG and RUNSTATS.

8. If you had a Business Process Choreographer Observer configuration, switch to the new reporting function by performing Enabling the Business Process Choreographer Explorer reporting function after migration.

9. If you have migrated from version 6.0.2, and 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 provides compatibility with earlier versions by maintaining the 6.0.2 behavior of not requiring a login when application security is enabled. But the use of the value Everyone is deprecated. 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:

   a. Open the administrative console and select **Applications > Enterprise Applications**.

   b. In the right panel, click on the name BPEContainer_*scope*, where *scope* is either *nodeName_serverName* or *clusterName*, depending on whether you configured Business Process Choreographer on a server or on a cluster.

c. In the right panel, under Detail Properties, select **Security role to user/group mapping**.

d. Change the mapping for the J2EE BPEAPIUser role from ″Everyone″ to ″All authenticated″.

e. Select **OK**.

f. Repeat these steps for the TaskAPIUser role of the TaskContainer_*name* enterprise application.

g. Save your changes, then restart the server or cluster on which you configured Business Process Choreographer.

10. If you modified the `faces-config-beans.xml` configuration file to specify thresholds for the queries for the Business Process Choreographer Explorer in version 6.0.2 or version 6.1.x before upgrading to version 6.2, you must re-apply the changes. For more information, refer to the following Technote: Business Process Choreographer Explorer - Customization and Tuning Options.

    Note: Since version 6.1, only predefined views are affected by the settings in the `faces-config-beans.xml` file. The thresholds for custom views are specified as part of their definition.

Related information

Uninstalling business process and human task applications, using the administrative console

Uninstalling business process and human task applications, using administrative commands

Postmigration tasks for Business Space powered by WebSphere

After migrating WebSphere Process Server from version 6.1.2 to version 6.2, you must perform some additional tasks before you start your servers or clusters.

Before you begin

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

About this task

If you are migrating from WebSphere Process Server version 6.1.2 and you have Business Space configured, you must perform two manual steps after migration before you can use Business Space.

Procedure

1. Upgrade the Business Space database. To do this, you run two scripts. For instructions, see ″Upgrading the Business Space database manually.″
2. Configure endpoints used by widgets in Business Space. You update the endpoints for widgets to appear in Business Space by using the REST Service Endpoints page in the administrative console. See ″Enabling Business Space widget endpoints on the administrative console.″

Results

You can use Business Space version 6.2.

Note: If you have used Business Space version 6.1.2, you must clear your browser cashe before using Business Space version 6.2. This will help you avoid inadvertent, continued use of code and images from Business Space version 6.1.2.

Related tasks

Upgrading the Business Space database manually
After migrating the server from version 6.1.2, you must manually upgrade the Business Space database to a new database schema before you start the version 6.2 server.

Enabling Business Space widget endpoints on the administrative console

Enabling Business Space widgets in the endpoints files

Migrating Cloudscape databases

After you use the migration tools to migrate to WebSphere Process Server version 6.2, you should verify the results of the automatic Cloudscape database migration and manually migrate any Cloudscape database instances that are not automatically migrated by the tools.

Before you begin

See "Overview of migrating" on page 1 and "Premigration considerations for WebSphere Process Server" on page 4.

Tips:

- 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.
- Before you run the migration tools, ensure that the *debug migration trace* is active. By default, this trace function is enabled. To reactivate the debug migration trace if it is disabled, set one of the following trace options:
  - `all traces*=all`
  - `com.ibm.ws.migration.WASUpgrade=all`

About this task

WebSphere Process Server version 6.2 requires Cloudscape Version 10.1.

Cloudscape Version 10.1 is a pure Java database server that combines the Apache Derby runtime with the opportunity to use the full services of IBM Software Support. For comprehensive information about Cloudscape Version 10.1, see the Cloudscape product web pages.

For help in troubleshooting problems when migrating, see "Troubleshooting version-to-version migration" on page 140.

Procedure

1. Verify the automatic migration of Cloudscape database instances.

   When you migrate from WebSphere Process Server version 6.1.x or 6.0.2.x to version 6.2, the migration tools automatically upgrade the database instances that are accessed through the embedded framework by some internal components such as the UDDI registry. The tools also attempt to upgrade

Cloudscape instances that your applications access through the embedded framework. You must verify these migration results after running the migration tools.

See “Verifying the Cloudscape v10.1.x automatic migration” on page 130.

2. Manually migrate Cloudscape database instances where necessary.

   The version 6.2 migration tools do not attempt to automatically migrate database instances that transact with applications through the Cloudscape Network Server framework. This exclusion eliminates the risk of corrupting third-party applications that access the same database instances as those accessed by WebSphere Process Server

   For details on manually migrating database instances that are accessed through the Cloudscape Network Server framework as well as Cloudscape instances that fail the automatic migration, see “Upgrading Cloudscape manually” on page 134.

3. Manually migrate your UDDI registry if it uses a database on the Cloudscape Network Server framework.

   See “Migrating the UDDI registry” on page 138.

Related concepts

How data is handled during migration from earlier versions
The WebSphere Process Server version-to-version migration tools will handle different sets of data (application data, configuration data, database information, and long-running processes) in different ways.

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

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

Related tasks

Upgrading databases for migration
In conjunction with migration, the database schema of some WebSphere Process Server components must be upgraded. This can occur automatically but in some cases you must upgrade the schema manually.

Verifying the Cloudscape v10.1.x automatic migration
WebSphere Process Server version 6.2 requires Cloudscape to run at a minimal version of v10.1.x. (Note that Cloudscape v10.1.x is comprised of the code base from Apache Derby Version 10.1.) During the WebSphere Process Server version 6.2 upgrade, the migration tool automatically upgrades the database instances that are accessed through the embedded framework by some internal components, such as the UDDI registry. The tool also attempts to upgrade Cloudscape instances that your applications access through the embedded framework. You must verify the migration results for these backend databases.

Migrating the UDDI registry
With most scenarios, migration of existing UDDI registries happens automatically when you migrate to the current level of WebSphere Process Server. However, if your existing UDDI registry uses a network Cloudscape database or a DB2 UDDI Version 2 database, there are some manual steps that you must take.

Related information

IBM Cloudscape product Web pages

Cloudscape migration document

Verifying the Cloudscape v10.1.x automatic migration

WebSphere Process Server version 6.2 requires Cloudscape to run at a minimal version of v10.1.x. (Note that Cloudscape v10.1.x is comprised of the code base from Apache Derby Version 10.1.) During the WebSphere Process Server version 6.2 upgrade, the migration tool automatically upgrades the database instances that are accessed through the embedded framework by some internal components, such as the UDDI registry. The tool also attempts to upgrade Cloudscape instances that your applications access through the embedded framework. You must verify the migration results for these backend databases.

Before you begin

Do not use Cloudscape v10.1.x as a production database. Use it for development and test purposes only.

Learn more: The new version of Cloudscape combines the Derby runtime with additional benefits, such as IBM Quality Assurance (QA) and national language support (NLS). For information about the Cloudscape v10.1.x open source code base, see the Cloudscape product Web pages.

The migration tool attempts to upgrade Cloudscape database instances that are accessed through the embedded framework only. You must manually upgrade Cloudscape instances that transact with servers on the Derby Network Server framework. (See "Upgrading Cloudscape manually" on page 134.) This requirement eliminates the risk of corrupting third party applications that use the Network Server framework to access the same database instances as WebSphere Process Server

Other applications can access Cloudscape on Network Server because the framework provides the database with a foundation of connectivity software; the embedded framework does not. Cloudscape Network Server can transact with multiple Java Virtual Machines (JVM)s (or servers) concurrently, whereas Cloudscape on the embedded framework works with only a single JVM. Clustered or coexistence implementations of WebSphere Process Server require Network Server. For more information, consult the IBM Cloudscape information center.

About this task

For database instances that your applications access through the embedded framework, the automatic migration can succeed completely, fail completely, or succeed with warnings. A migration that produces warning messages does create a Cloudscape v10.1.x database with your data, but does not migrate all of your configured logic and other settings, such as:

- keys
- checks
- views
- triggers
- aliases
- stored procedures

To distinguish between a partially and a completely successful migration, you must verify the auto-migration results by checking both the general post-upgrade log and the individual database logs. Performing these tasks gives you vital diagnostic data to troubleshoot the partially migrated databases as well as those that fail auto-migration completely. Ultimately, you migrate these databases through a manual process.

Procedure

1. Open the post-upgrade log of each new WebSphere Process Server version 6.2 profile. The path name of the log is `*install_root*profiles/*profileName*/logs/WASPostUpgrade.*timestamp*.log`.
2. Examine the post-upgrade log for database error messages. These exceptions indicate database migration failures. The following lines are an example of post-upgrade log content, in which the database error code is DSRA7600E. (The migration tool references all database exceptions with the prefix DSRA.)

   ```
   MIGR0344I: Processing configuration file /opt/WebSphere60/AppServer/cloudscape
   /db2j.properties.

   MIGR0344I: Processing configuration file /opt/WebSphere60/AppServer/config/cells
   /migr06/applications/MyBankApp.ear/deployments/MyBankApp/deployment.xml.

   DSRA7600E: Cloudscape migration of database instance /opt/WebSphere61/Express
   /profiles/default/databases/_opt_WebSphere60_AppServer_bin_DefaultDB failed,
   ```

```
reason: java.sql.SQLException: Failure creating target db

MIGR0430W: Cloudscape Database /fvt/temp/60BaseXExpress/PostUpgrade50BaseFVTTest9
/testRun/pre/websphere_backup/bin/DefaultDB failed to migrate <new database name>
```

Important: Call IBM WebSphere Process Server Support if you see a migration failure message for a Cloudscape instance that is accessed by a WebSphere internal component (that is, a component that helps comprise WebSphere Process Server rather than one of your applications).

3. Open the individual database migration log that corresponds with each of your backend Cloudscape databases. These logs have the same timestamp as that of the general post-upgrade log. The logs display more detail about errors that are listed in the general post-upgrade log, as well as expose errors that are not documented by the general log.

   The path name of each database log is WAS_HOME/profiles/*profileName*/logs/*myFulldbPathName*_migrationLog*timestamp*.log.

4. Examine each database migration log for errors. For a completely successful migration, the log displays a message that is similar to the following text:

   ```
   MIGR0429I: Cloudscape Database F:\temp\60BaseXExpress\PostUpgrade50BaseFVTTest2\testRun
   \pre\websphere_backup\bin\DefaultDB was successfully migrated. See log C:\WebSphere61
   \Express\profiles\default\logs\DefaultDB_migrationLogSun-Dec-18-13.31.40-CST-2005.log
   ```

   Otherwise, the log displays error messages in the format of the following example:

   ```
   connecting to source db <jdbc:db2j:/fvt/temp/60BaseXExpress/PostUpgrade50BaseFVTTest9
   /testRun/pre/websphere_backup/bin/DefaultDB>

   connecting to source db <jdbc:db2j:/fvt/temp/60BaseXExpress/PostUpgrade50BaseFVTTest9
   /testRun/pre/websphere_backup/bin/DefaultDB> took 0.26 seconds

   creating target db <jdbc:derby:/opt/WebSphere61/Express/profiles/default/databases
   /_opt_WebSphere60_AppServer_bin_DefaultDB>

    ERROR: An error occurred during migration. See debug.log for more details.

   shutting down databases

   shutting down databases took   0.055 seconds
   ```

5. For more data about a migration error, consult the debug log that corresponds with the database migration log. The WebSphere Application Server migration utility triggers a *debug migration trace* by default; this trace function generates the database debug logs. The full path name of a debug log is WAS_HOME/profiles/*profileName*/logs/*myFulldbPathName*_migrationDebug*timestamp*.log.

   The following lines are a sample of debug text. The lines display detailed exception data for the error that is referenced in the previous sample of database migration log data.

   ```
   java.sql.SQLException: Database_opt_WebSphere60_AppServer_bin_DefaultDB already exists.
   Aborting migration
    at com.ibm.db2j.tools.migration.MigrateFrom60Impl.go(Unknown Source)
    at com.ibm.db2j.tools.migration.MigrateFrom60Impl.doMigrate(Unknown Source)
    at com.ibm.db2j.tools.MigrateFrom60.doMigrate(Unknown Source)
    at com.ibm.ws.adapter.migration.CloudscapeMigrationUtility.migr
   ```

Results

- The WebSphere Process Server migration utility changes your Cloudscape JDBC configurations whether or not it successfully migrates the database instances that are accessed by your applications. The tool changes Cloudscape JDBC provider class paths, data source implementation classes, and data source helper classes. The following table depicts these changes:

Table 5. New class information

Class type	Old value	New value
JDBC provider class path	`${CLOUDSCAPE_JDBC_DRIVER_PATH}/db2j.jar`	`${DERBY_JDBC_DRIVER_PATH}/derby.jar` • Where `DERBY_JDBC_DRIVER_PATH` is the WebSphere environment variable that defines your Cloudscape JDBC provider • Where `derby.jar` is the base name of the JDBC driver class file (In your environment, reference the JDBC driver class file by the full path name.)
Data source implementation class: Connection pool	`com.ibm.db2j.jdbc.DB2jConnectionPool DataSource`	`org.apache.derby.jdbc.EmbeddedConnection PoolDataSource`
Data source implementation class: XA	`com.ibm.db2j.jdbc.DB2jXADataSource`	`org.apache.derby.jdbc.EmbeddedXADataSource`
Data source helper class	`com.ibm.websphere.rsadapter.Cloudscape DataStoreHelper`	`com.ibm.websphere.rsadapter.Derby DataStoreHelper`

Additionally, the `db2j.properties` file changes:

- The name `WAS_HOME/cloudscape/dbj.properties` changes to `WAS_HOME/derby/derby.properties`
- Within the file, property names change from `db2j.drda.*` to `derby.drda.*`

- A partial or a completely successful database migration changes the location and name of the database according to the following example:
  - **Old database name:** `c:\temp\`*`mydb`*
  - **New database name:** The new name includes a hash code that combines the entire path name of the old database and the migration time stamp. The new name also includes the old database name and time stamp verbatim. Example: *`install_root`*`\profiles\profile_name\databases\`*`my_db`*`_hashCode_timestamp`

Note the exact path names: For both partial and failed migrations, the log messages contain the exact old and new database path names that you must use to run the manual migration. Note these new path names precisely.

What to do next

If you experience a partial migration, attempt to troubleshoot the new v10.1.x database only if you have expert knowledge of Cloudscape. Otherwise, delete the new database. Perform the manual migration procedure on the original database, just as you do for each database that completely fails auto-migration. Consult "Upgrading Cloudscape manually" on page 134 for instructions.

For successfully migrated Cloudscape instances, be aware that new cell-scoped data sources can only be used by nodes that run version 6.0.2 or later of WebSphere Process Server. Earlier versions of the product do not support the new Cloudscape; when applications on pre-version 6.0.2 nodes try to access a Cloudscape 10.1.x data source, the server will issue exceptions at run time.

Related tasks

Upgrading Cloudscape manually
During the WebSphere Process Server version 6.2 upgrade, the migration tools attempt to upgrade instances of Cloudscape that are accessed through the embedded framework only. (The new version of Cloudscape is version 10.1.x, which is based on Derby.) The automatic upgrade excludes Cloudscape instances that transact with applications through the Network Server framework. This exclusion eliminates the risk of corrupting third party applications that access the same database instances as WebSphere Process Server. You must manually upgrade database instances that are accessed through the Network Server framework. Do the same for databases that fail the automatic migration.

Migrating Cloudscape databases
After you use the migration tools to migrate to WebSphere Process Server version 6.2, you should verify the results of the automatic Cloudscape database migration and manually migrate any Cloudscape database instances that are not automatically migrated by the tools.

Migrating the UDDI registry
With most scenarios, migration of existing UDDI registries happens automatically when you migrate to the current level of WebSphere Process Server. However, if your existing UDDI registry uses a network Cloudscape database or a DB2 UDDI Version 2 database, there are some manual steps that you must take.

Related information

IBM Cloudscape product Web pages

Cloudscape migration document

Apache Derby

IBM Cloudscape information center

Upgrading Cloudscape manually

During the WebSphere Process Server version 6.2 upgrade, the migration tools attempt to upgrade instances of Cloudscape that are accessed through the embedded framework only. (The new version of Cloudscape is version 10.1.x, which is based on Derby.) The automatic upgrade excludes Cloudscape instances that transact with applications through the Network Server framework. This exclusion eliminates the risk of corrupting third party applications that access the same database instances as WebSphere Process Server. You must manually upgrade database instances that are accessed through the Network Server framework. Do the same for databases that fail the automatic migration.

Before you begin

Do not use Cloudscape v10.1.x as a production database. Use it for development and test purposes only.

Learn more: The new version of Cloudscape combines the Derby runtime with additional benefits, such as IBM Quality Assurance (QA) and national language support (NLS).

- For information about the Cloudscape v10.1.x open source code base, see the Cloudscape product Web pages.
- For information about incompatibilities between Cloudscape v10.1.x and v5.1.60x (plus versions prior to v5.1.60x) see Migrating IBM Cloudscape to Version 10.

For instances of Cloudscape that are accessed through the embedded framework, determine which instances completely failed the automatic upgrade process and which ones were only partially upgraded. The topic “Verifying the Cloudscape v10.1.x automatic migration” on page 130 documents how to uncover database errors and diagnostic data from various migration logs. The log messages contain the exact old and new database path names that you must use to run the manual migration. Note these new path names precisely.

To minimize the risk of migration errors for databases that were only partially upgraded during the automatic migration process, delete the new database. Troubleshoot the original database according to the log diagnostic data, then perform manual migration on the original database.

About this task

The following section consists of steps to migrate Cloudscape instances that are accessed through both frameworks: the embedded as well as the Network Server framework. Steps that apply only to the Cloudscape Network Server framework are marked accordingly. As a migration best practice, ensure that your user ID has one of the following authorities:

- Administrator of the server that accesses the Cloudscape instance
- A umask that can access the database instance

Otherwise, you might see runtime errors about the database instance being read-only.

Procedure

1. **Network Server framework only:** Ensure that every client of the Cloudscape database can support Cloudscape v10.1.x. WebSphere Process Server clients of the database must run versions 6.0.1.x or higher of WebSphere Process Server.
2. **Network Server framework only:** Take the database offline. No clients can access it during the migration process.
3. Examine a sample Cloudscape migration script that WebSphere Process Server provides. Depending on your operating system, WebSphere Process Server provides one of the following migration scripts:
   - Linux UNIX **On Linux and UNIX platforms**: Use the `db2jmigrate.sh` script, located in the following directory: *install_root*/derby/bin/embedded/...
   - Windows **On Windows platforms**: Use the `db2jmigrate.bat` script, located in the following directory: *install_root*\derby\bin\embedded\...

   You can modify the script according to the requirements of your environment. Consult Migrating IBM Cloudscape to Version 10 for information about options that you can use with the script. For example, you can use the option `-DB2j.migrate.ddlFile=`*`filename`* to specify the DDL file for the new database.
4. To generate database debug logs when you run the migration script, ensure that the *debug migration trace* is active. By default, this trace function is enabled. Reactivate the debug trace if it is disabled.
   a. To set the trace options in the administrative console, click **Troubleshooting** > **Logging and Tracing** in the console navigation tree.
   b. Select the server name.
   c. Click **Change Log Level Details**.

d. Optional: If **All Components** has been enabled, you might want to turn it off, and then enable specific components.

e. Optional: Select a component or group name. For more information see Log level settings in the WebSphere Application Server Network Deployment, version 6.1 Information Center. If the selected server is not running, you will not be able to see individual component in graphic mode.

f. Enter a trace string in the trace string box. In this case, enter one of the following:

   - all traces*=all
   - com.ibm.ws.migration.WASUpgrade=all

   For more information on tracing read Working with trace in the WebSphere Application Server Network Deployment, version 6.1 Information Center..

g. Select **Apply**, then **OK**.

5. Specify your old database name and the full postmigration path of the new database name when you run the script. For example: `E:\WebSphere\ProcServer\derby\bin\embedded>db2jMigrate.bat` *`myOldDB myNewDB`* The logs from the automatic migration provide the exact path names to specify for both the old database and the target database. You must use this target database name to specify the new database, because your migrated Cloudscape data sources (updated by the WebSphere Process Server migration utilities) now point to the target database name. The following sample text demonstrates how log messages display target database names:

   ```
   Cloudscape migration of database instance C:\temp\migration2\profiles\Srv01\
   installedApps\ghongellNode01Cell\DynamicQuery.ear\EmployeeFinderDB to
   new database instance C:\WebSphere\ProcServer
   \profiles\Srv01\databases\C__WAS602_ProcServer_profiles_ProcSrv01_
   installedApps_ghongellNode01Cell_DynamicQuery.ear_
   EmployeeFinderDB failed, reason: java.sql.SQLException:
   Failure creating target db
   ```

   For instances of Cloudscape that are accessed through the Network Server framework, input any name that you want for the new database. Remember to modify your existing data sources to point to the new database name.

6. When the migration process ends, examine the database migration log to verify the results. The path name of each database migration log is *`install_root`*`/logs/derby/`*`myFulldbPathName`*`_migrationLog.log`.

   For a successful migration, the database migration log displays a message that is similar to the following text:

   ```
   Check E:\WebSphere\ProcServer\derby\myOldDB_migrationLog.log for progress
   Migration Completed Successfully
   E:\WebSphere\ProcServer\derby\bin\embedded>
   ```

   Otherwise, the log displays error messages in the format of the following example:

   ```
   Check E:\WebSphere\ProcServer\derby\myOldDB_migrationLog.log for progress
   ERROR: An error occurred during migration. See debug.log for more details.
   ERROR XMG02: Failure creating target db
   java.sql.SQLException: Failure creating target db
       at com.ibm.db2j.tools.migration.MigrationState.getCurrSQLException(Unknown
       Source)
       at com.ibm.db2j.tools.migration.MigrateFrom51Impl.handleException(Unknown
       Source)
       at com.ibm.db2j.tools.migration.MigrateFrom51Impl.go(Unknown Source)
       at com.ibm.db2j.tools.migration.MigrateFrom51Impl.main(Unknown Source)
       at com.ibm.db2j.tools.MigrateFrom51.main(Unknown Source)
   ```

7. For more data about a migration error, consult the debug log that corresponds with the database migration log. The full path name of a debug log file is `install_root/logs/derby/myFulldbPathName_migrationDebug.log`

   The following lines are a sample of debug text.

   ```
   sourceDBURL=jdbc:db2j:E:\WebSphere\myOldDB
    newDBURL=jdbc:derby:e:\tempo\myNewDB
    ddlOnly=false
   connecting to source db <jdbc:db2j:E:\WebSphere\myOldDB>
   connecting to source db <jdbc:db2j:E:\WebSphere\myOldDB> took   0.611 seconds
   creating target db <jdbc:derby:e:\tempo\myNewDB>
   creating target db <jdbc:derby:e:\tempo\myNewDB> took   6.589 seconds
   initializing source db data structures
   initializing source db data structures took   0.151 seconds
   recording DDL to create db <E:\WebSphere\myOldDB>
   recording DDL to create db <E:\WebSphere\myOldDB> took   5.808 seconds
   ```

Results

As indicated in the previous steps, the database migration log displays either a `Migration Completed Successfully` message, or a message containing migration failure exceptions.

What to do next

- For databases that fail migration, troubleshoot according to the logged error data. Then rerun the migration script.
- To access successfully upgraded databases through the embedded framework, modify your data sources to point to the new database names.
- To access successfully upgraded databases through the Network Server framework, you can use either the DB2 Universal JDBC driver or the Derby Client JDBC driver.
  - If you want your existing JDBC configurations to continue to use the DB2 Universal JDBC driver, modify your data sources to point to the new database names.
  - If you want to use the Derby Client JDBC driver, which can support XA data sources, modify your JDBC providers to use the new Derby Client JDBC driver class and the new data source implementation classes. Then reconfigure every existing data source to use the correct Derby data source helper class, and to point to the new database name.

    Consult the topic Data source minimum required settings, by vendor in the WebSphere Application Server Network Deployment, version 6.1 Information Center for all of the new class names.
  - Run the database upgrade scripts in the *install_root*/dbscripts/*component_name*/Derby directory to upgrade the database tables and schema to the WebSphere Process Server version 6.2 level. For more information see "Upgrading databases for migration" on page 34.

Related tasks

Verifying the Cloudscape v10.1.x automatic migration
WebSphere Process Server version 6.2 requires Cloudscape to run at a minimal version of v10.1.x. (Note that Cloudscape v10.1.x is comprised of the code base from Apache Derby Version 10.1.) During the WebSphere Process Server version 6.2 upgrade, the migration tool automatically upgrades the database instances that are accessed through the embedded framework by some internal components, such as the UDDI registry. The tool also attempts to upgrade Cloudscape instances that your applications access through the embedded framework. You must verify the migration results for these backend databases.

Migrating the UDDI registry
With most scenarios, migration of existing UDDI registries happens automatically when you migrate to the current level of WebSphere Process Server. However, if your existing UDDI registry uses a network Cloudscape database or a DB2 UDDI Version 2 database, there are some manual steps that you must take.

Upgrading databases for migration
In conjunction with migration, the database schema of some WebSphere Process Server components must be upgraded. This can occur automatically but in some cases you must upgrade the schema manually.

Related information

IBM Cloudscape product Web pages

Cloudscape migration document

Migrating IBM Cloudscape to Version 10

Log level settings

Working with trace

Data source minimum required settings, by vendor

Migrating the UDDI registry

With most scenarios, migration of existing UDDI registries happens automatically when you migrate to the current level of WebSphere Process Server. However, if your existing UDDI registry uses a network Cloudscape database or a DB2 UDDI Version 2 database, there are some manual steps that you must take.

Before you begin

Migrate your installation of WebSphere Process Server; ensure that you select the option to migrate applications, so that the UDDI registry application will be migrated.

About this task

If your existing UDDI registry uses an Oracle, embedded Cloudscape or DB2 UDDI Version 3 database, you do not need to perform any manual migration; migration happens automatically when you migrate WebSphere Process Server and start the UDDI node for the first time after migration.

If your existing UDDI registry uses a network Cloudscape database or a DB2 UDDI Version 2 database, you must complete some manual steps to migrate the registry.

- If your UDDI registry uses a DB2 UDDI Version 2 database, follow the steps in Migrating to Version 3 of the UDDI registry. and sub-topics.
- If your UDDI registry uses a network Cloudscape database, complete the following steps.
  1. If you have a cluster that contains servers at different levels of WebSphere Process Server, ensure that any UDDI registries are running on servers that are at WebSphere Process Server version 6.2. For example, if you have a cluster that spans two nodes, you can upgrade one node to WebSphere Process Server version 6.2 while the other node remains at a previous level, provided that any servers that are running a UDDI registry are at version 6.2.
  2. Initialize the relevant UDDI node. The initialize process will perform some of the UDDI registry migration.
  3. Enter the following commands as the database administrator, from *install_root*/cloudscape/lib.

     ```
     java -cp db2j.jar;db2jtools.jar com.ibm.db2j.tools.ij

     connect 'jdbc:db2j:uddi_cloudscape_database_path';

     run 'install_root/UDDIReg/databaseScripts/uddi30crt_drop_triggers
     _cloudscape.sql';

     quit;

     cd install_root/derby/migration

     java -cp db2j.jar;db2jmigration.jar;../lib/derby.jar
     com.ibm.db2j.tools.MigrateFrom51
        jdbc:db2j:uddi_cloudscape_database_path
     ```

     where

     – *uddi_cloudscape_database_path* is the absolute path of the existing Cloudscape database, for example *install_root*/profiles/*profile_name*/ databases/com.ibm.uddi/UDDI30

     – *install_root* is the root directory for the installation of WebSphere Process Server.

Results

The UDDI database and data source are migrated, and the UDDI node is activated.

Note: When you migrate WebSphere Process Server, the post-upgrade log for the profile indicates that the migration of the UDDI database is partially complete, and is missing the steps for triggers, aliases, and stored statements. If you initially enabled the debug function, the debug log for the database indicates that there was a failure creating triggers. Ignore these messages; the UDDI node completes the migration of the database when the UDDI node starts. For more information about these log files, see “Verifying the Cloudscape v10.1.x automatic migration” on page 130. Also refer to this topic if other errors appear in the logs.

If the migration of the UDDI database completes successfully, the following message appears in the server log:

```
CWUDQ0003I: UDDI registry migration has completed
```

If the following error appears, an unexpected error occurred during migration. The UDDI registry node is not activated. Check the error logs for the problem and, if it cannot be fixed, see the IBM software support Web site at http://www.ibm.com/software/support.

```
CWUDQ004W: UDDI registry not started due to migration errors
```

Related tasks

Verifying the Cloudscape v10.1.x automatic migration
WebSphere Process Server version 6.2 requires Cloudscape to run at a minimal version of v10.1.x. (Note that Cloudscape v10.1.x is comprised of the code base from Apache Derby Version 10.1.) During the WebSphere Process Server version 6.2 upgrade, the migration tool automatically upgrades the database instances that are accessed through the embedded framework by some internal components, such as the UDDI registry. The tool also attempts to upgrade Cloudscape instances that your applications access through the embedded framework. You must verify the migration results for these backend databases.

Upgrading Cloudscape manually
During the WebSphere Process Server version 6.2 upgrade, the migration tools attempt to upgrade instances of Cloudscape that are accessed through the embedded framework only. (The new version of Cloudscape is version 10.1.x, which is based on Derby.) The automatic upgrade excludes Cloudscape instances that transact with applications through the Network Server framework. This exclusion eliminates the risk of corrupting third party applications that access the same database instances as WebSphere Process Server. You must manually upgrade database instances that are accessed through the Network Server framework. Do the same for databases that fail the automatic migration.

Migrating Cloudscape databases
After you use the migration tools to migrate to WebSphere Process Server version 6.2, you should verify the results of the automatic Cloudscape database migration and manually migrate any Cloudscape database instances that are not automatically migrated by the tools.

Related information

Migrating to Version 3 of the UDDI registry

IBM Software Support web site

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

The following sections describe specific errors and exceptions that may occur in a version-to-version migration and provide steps you can follow to understand and resolve these problems.

- “Application installation error” on page 141
- “Application server error” on page 141
- “Exceptions: database connectivity, loading, or missing class” on page 142
- “Out of memory error” on page 142
- “Profile creation error” on page 143
- “Profile migration error” on page 143
- “Servlet error” on page 144
- “Synchronization error” on page 144

Application installation error

If you select the option for the migration process to install the enterprise applications that exist in the version 6.1.x or 6.0.2.x configuration into the new version 6.2 configuration, you might encounter some error messages during the application-installation phase of migration.

The applications that exist in the version 6.1.x or 6.0.2.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.

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

- Fix the problems in the version 6.1.x or 6.0.2.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 6.2 configuration using the administrative console or an install script.

Application server error

After you migrate a managed node to version 6.2, 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.

Exceptions: database connectivity, loading, or missing class

Never change any WebSphere Application Server variables that are configured as a part of profile creation.

If you modify these values incorrectly in old profile, you might get database connectivity, loading, or other missing class exceptions, such as:

```
10/25/08 13:22:39:650 GMT+08:00] 0000002e J2CUtilityCla E   J2CA0036E: An
exception occurred while invoking method setDataSourceProperties on
com.ibm.ws.rsadapter.spi.WSManagedConnectionFactoryImpl used by resource
jdbc/com.ibm.ws.sib/ewps6101.Messaging-BPC.cwfpcCell01.Bus :
com.ibm.ws.exception.WsException: DSRA0023E: The DataSource implementation
class "com.ibm.db2.jcc.DB2XADataSource" could not be found.DB2,
```

Derby, and SQL Embedded JDBC drivers are bundled with the WebSphere Process Server product installation. If you need to change these drivers to any higher version, you must copy drivers on same location where they exists in the product installation, as follows:

- **Derby**: %*was.install.root*%\derby\lib
- **DB2**: %*was.install.root*%/universalDriver_wbi/lib
- **SQL**: %*was.install.root*%lib

If you need a new JDBC provider and datasource for your application, you can create these resources by selecting a valid jdbcclasspath and setting the WebSphere Application Server variable accordingly. For example, if you need DB2 at cell level which doesn't exist earlier in your installation, you could use the following procedure.

1. In the administrative console, navigate to: **Resources** → **JDBC** → **JDBC Providers** → **DB2 Universal JDBC Driver Provider (XA)**.
2. In the **Class path** box, set the following paths:
   - DB2UNIVERSAL_JDBC_DRIVER_PATH =%was.install.root%/universalDriver_wbi/lib
   - DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH=""

   If you need your own drivers, set the following path: DB2UNIVERSAL_JDBC_DRIVER_PATH=%myDriverLocation%

Out of memory error

If either the WBIPreUpgrade or WBIPostUpgrade command-line utility fail due to Out of Memory problems, you can increase the heap size to a number that takes into consideration the size and scope of the environment being migrated, as well as what the machine will allow.

For instructions on how to increase the heap size, use the procedure described in Solution 4 of the following technote: Instructions for handling certain Out of Memory conditions.

Profile creation error

While you are using the version 6.2 migration wizard to create a profile when migrating a configuration, you might see the following profile-creation error messages.

```
profileName: profileName cannot be empty
profilePath: Insufficient disk space
```

These error messages might be displayed if you enter a profile name that contains an incorrect character such as a space. Rerun the migration wizard, and verify that there are no incorrect characters in the profile name such as a space, quotes, or any other special characters.

Profile migration error

When you use the migration wizard to migrate a profile from WebSphere Process Server version 6.1.x or 6.0.2.x to version 6.2 on a Solaris x64 processor-based system, the migration might fail during the WBIPostUpgrade step.

You might see messages similar to the following in *profile_root*/logs/ WASPostUpgrade.*time_stamp*.log:

```
MIGR0327E: A failure occurred with stopNode.
MIGR0272E: The migration function cannot complete the command.
```

WebSphere Process Server version 6.1.x or 6.0.2.x uses a Java virtual machine (JVM) in 32-bit mode. The migration wizard for WebSphere Process Server version 6.2 calls the WBIPostUpgrade.sh script, which attempts to run the JVM for version 6.1.x or 6.0.2.x in the 64-bit mode when the server stops the version 6.1.x or 6.0.2.x node.

Complete the following actions to remove the incomplete profile and enable WebSphere Process Server to correctly migrate the version 6.1.x or 6.0.2.x profile:

1. On a command line, change to the *install_root*/bin directory.

   For example, type the following command:

   ```
   cd /opt/IBM/WebSphere/Procserver/bin
   ```
2. Locate the WBIPostUpgrade.sh script in the *install_root*/bin directory, and make a backup copy.
3. Open the WBIPostUpgrade.sh or WBIPostUpgrade.bat file in an editor, and perform the following actions:
   1. Locate the following line of code:

      UNIX Linux

      ```
      "$binDir" /setupCmdLine.sh
      ```

      Windows

      ```
      call "%~dp0setupCmdLine.bat" %*
      ```
   2. Insert the following line of code after the code that was identified in the previous step:

      ```
      JVM_EXTRA_CMD_ARGS=""
      ```
   3. Save the changes.
4. Repeat steps 2 through 4 with theWASPostUpgrade.sh or the WASPostUpgrade.bat file.
5. Delete the incomplete version 6.2 profile that was created during the migration process. Use the following procedure.

a. Open a command prompt and run one of the following commands, based on your operating system:
   - i5/OS **On i5/OS platforms:** `manageprofiles -delete -profileName` *`profile_name`*
   - Linux UNIX **On Linux and UNIX platforms:** `manageprofiles.sh -delete -profileName` *`profile_name`*
   - Windows **On Windows platforms:** `manageprofiles.bat -delete -profileName` *`profile_name`*

   The variable *profile_name* represents the name of the profile that you want to delete.

b. Confirm that the profile deletion has completed by checking the following log file:
   - i5/OS **On i5/OS platforms:** *user_data_root*/profileRegistry/logs/manageprofiles/*profile_name*_delete.log
   - Linux UNIX **On Linux and UNIX platforms:** *install_root*/logs/manageprofiles/*profile_name*_delete.log
   - Windows **On Windows platforms:** *install_root*\logs\manageprofiles\profile_name_delete.log

6. Delete the *profile_root* directory of the version 6.2 profile that was removed in the previous step.
7. Rerun the migration wizard.

Servlet error

In a network deployment environment, if the error `SRVE0026E: [Servlet Error]-[com/ibm/wbiservers/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.

Synchronization error

If synchronization fails when you migrate a managed node to version 6.2, the server might not start.

You might receive messages similar to the following when you migrate a managed node to version 6.2:

```
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/62AppServer/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 6.2 configuration level.

- The managed node that you are trying to migrate is at a version 6.2 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 .

Related concepts

Premigration considerations for Business Process Choreographer
If your servers run Business Process Choreographer, you should be aware of certain things that you need to plan and take into consideration before you migrate Business Process Choreographer.

Related tasks

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

Related reference

WBIPreUpgrade command-line utility
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-line utility
Use the WBIPostUpgrade command for WebSphere Process Server to retrieve the profile configuration that was saved by the WBIPreUpgrade command at the *backupDirectory* that you had specified.

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 other 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 products:

- WebSphere InterChange Server version 4.2.0 or later. For more information see “Migrating from WebSphere InterChange Server or WebSphere Business Integration Server Express.”
- WebSphere Business Integration Server Foundation versions 5.1 and 5.1.1. For more information see “Migrating from WebSphere Studio Application Developer Integration Edition” on page 201.
- WebSphere MQ Workflow version 3.6. For more information, see “Migrating from WebSphere MQ Workflow” on page 202.

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.

i5/OS Although these products were not supported on i5/OS, modules from them can be migrated to WebSphere Process Server version 6.2 on their respective platforms (using the migration tools available such as the reposMigrate command) and then deployed to WebSphere Process Server version 6.2 running on the i5/OS operating system.

For migration from another product to WebSphere Process Server (for example, WebSphere InterChange Server to WebSphere Process Server) , 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 to 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 from WebSphere InterChange Server or WebSphere Business Integration Server Express

Use the WebSphere Integration Developer wizard or the WebSphere Process Server reposMigrate command to migrate from WebSphere InterChange Server version 4.3 or later or WebSphere Business Integration Server Express version 4.4 or later to WebSphere Process Server 6.2.

About this task

For this version of WebSphere InterChange Server or WebSphere Business Integration Server Express...	Do this
WebSphere InterChange Server version 4.3 or later or WebSphere Business Integration Server Express version 4.4 or later	Use the Migration wizard from WebSphere Integration Developer to migrate WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts into WebSphere Process Server deployable artifacts and place them into projects in the active WebSphere Integration Developer workspace. Alternatively, you can use the reposMigrate command to migrate WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts into WebSphere Process Server deployable artifacts, and optionally deploy them directly to WebSphere Process Server.
WebSphere InterChange Server versions earlier than 4.3 or WebSphere Business Integration Server Express versions earlier than 4.4	First migrate to WebSphere InterChange Server version 4.3 or later or WebSphere Business Integration Server Express version 4.4 or later, and then migrate to WebSphere Process Server.

Related information

Migrating WebSphere InterChange Server using the Migration wizard

WebSphere Integration Developer information center

Premigration considerations

Consider these guidelines for the development of integration artifacts for WebSphere InterChange Server or WebSphere Business Integration Server Express to ease the migration of WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts to WebSphere Process Server.

These recommendations are meant to be used only as a guide. There may be cases where it is necessary to deviate from these guidelines. In these cases care should be taken to limit the scope of the deviation to minimize the amount of rework required to migrate the artifacts. Note that the guidelines outlined here are not all general recommendations for the development of WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts. They are instead limited in scope to those considerations which may affect the ease in which artifacts can be migrated at a future time.

Related concepts

Troubleshooting migration from WebSphere InterChange Server or WebSphere Business Integration Server Express
Find solutions to problems you encounter with migration as well as instructions for turning on logging and tracing.

Related reference

Postmigration considerations
When applications have been migrated from WebSphere InterChange Server or WebSphere Business Integration Server Express to WebSphere Process Server, special attention is required in some areas to enable migrated applications to function in WebSphere Process Server consistently with their intended function due to differences between the architectures of WebSphere Process Server and WebSphere InterChange Server or WebSphere Business Integration Server Express.

Premigration considerations: Access framework clients

Do not develop any new clients adopting the CORBA IDL interface APIs. This is not supported in WebSphere Process Server.

Premigration considerations: Business objects

For the development of business objects, use only the tooling provided to configure artifacts, use explicit data types and lengths for data attributes, and use only the documented APIs.

Business objects within WebSphere Process Server are based on Service Data Objects (SDOs). SDOs use data attributes that are strongly typed. For business objects in WebSphere InterChange Server or WebSphere Business Integration Server Express and adapters, data attributes are not strongly typed, and users sometimes specify string data types for non-string data attributes. To avoid issues in WebSphere Process Server, explicitly specify data types.

Because business objects within WebSphere Process Server might be serialized at runtime as they are passed between components, it is important to be explicit with the required lengths for data attributes to minimize utilization of system resources. For this reason, do not use the maximum 255 character length for a string attribute, for example. Also, do not specify zero length attributes which currently default to 255 characters. Instead, specify the exact length required for attributes.

XSD NCName rules apply to business object attribute names in WebSphere Process Server. Therefore, do not use any spaces or ″:″ in names for business object attributes. Business object attribute names with spaces or ″:″ are invalid in WebSphere Process Server. Rename business object attributes before migration.

If using an array in a business object, you cannot rely on the order of the array when indexing into the array in Maps and/or Relationships. The construct that this migrates into in WebSphere Process Server does not guarantee index order, particularly when entries are deleted.

It is important to use only the Business Object Designer or the Business Object Designer Express™ tool to edit business object definitions, and to use only the published APIs for business objects within integration artifacts.

Premigration considerations: Collaboration templates

When developing WebSphere InterChange Server or WebSphere Business Integration Server Express collaboration templates, follow these guidelines to ensure the best chance of a smooth migration to WebSphere Process Server.

To ensure processes are described appropriately with metadata, always use the Process Designer tool for the creation and modification of collaboration templates, and avoid editing the metadata files directly. Use the Activity Editor tool wherever possible to maximize the use of metadata to describe the required logic.

To minimize the amount of manual rework that may be required in migration, use only the documented APIs within collaboration templates. Avoid the use of static variables. Instead, use non-static variables and collaboration properties to address the requirements of the business logic. Avoid the use of Java qualifiers final, transient and native in Java snippets. These cannot be enforced in the BPEL Java snippets that are the result of migrating the Collaboration Templates.

To maximize future portability, avoid using explicit connection release calls and explicit transaction bracketing (that is, explicit commits and explicit rollbacks) for User Defined Database Connection Pools. Instead, make use of the container-managed implicit connection clean-up and implicit transaction bracketing. Also, avoid keeping system connections and transactions active across Java snippet nodes within a collaboration template. This applies to any connection to an external system, as well as user-defined database connection pools. Operations with an external EIS should be managed within an adapter, and code related to database operation should be contained within one code snippet. This may be necessary within a collaboration which, when rendered as a BPEL business process component may be selectively deployed as an interruptible flow. In this case, the process may be comprised of several separate transactions, with only state and global variable information passed between the activities. The context for any system connection or related transaction which spanned these process transactions would be lost.

Name collaboration template property names in accordance with W3C XML NCName naming conventions. WebSphere Process Server accepts names conforming to those conventions. Any disallowed characters are invalid in BPEL property names that they will be migrated into. Rename properties to remove any disallowed characters before migrating to avoid syntactical errors in the BPEL generated by migration.

Do not reference variables using ″this.″ For example, Instead of ″this.inputBusObj″ use just ″inputBusObj″.

Use class-level scoping on variables instead of scenario-scoped variables. Scenario-scoping is not carried forward during migration.

Initialize all variables declared in Java snippets with a default value: ″Object myObject = null;″ for example. Be sure all variables are initialized during declaration before migrating.

Ensure that there are no Java import statements in the user modifiable sections of your collaboration templates. In the definition of the collaboration template, use the import fields to specify Java packages to import.

Do not set incoming business object values to be stored in the *triggeringBusObj* variable. Within WebSphere InterChange Server or WebSphere Business Integration Server Express, the *triggeringBusObj* is read-only and its values cannot be overwritten, so any incoming business object values will not be saved. If the *triggeringBusObj* is used as the receiving variable for an incoming business object on an inbound service call, then after migration the behavior of the inbound service call will be different: within the BPEL process, the incoming value from the inbound service call will overwrite the value stored in *triggeringBusObj*.

Premigration considerations: Common code utilities

IBM recommends that you avoid the development of common code utility libraries for use across integration artifacts within the WebSphere InterChange Server or WebSphere Business Integration Server Express environment. Consider using EJBs running in WebSphere Application Server to encapsulate the logic, and use web service calls to invoke them from WebSphere InterChange Server or WebSphere Business Integration Server Express.

While it is possible that some common code utility libraries may run appropriately on WebSphere Process Server, you will be responsible for the migration of the custom utilities.

Premigration considerations: Database connection pools

A WebSphere InterChange Server or WebSphere Business Integration Server Express database connection pool within a map or collaboration template will be rendered as a standard JDBC resource in WebSphere Process Server. However, the way connections and transactions are managed might differ between WebSphere InterChange Server or WebSphere Business Integration Server Express and WebSphere Process Server, so you should avoid keeping database transactions active across Java snippets.

User-defined database connection pools are useful within maps and collaboration templates for simple data lookups and for more sophisticated state management across process instances. A database connection pool in WebSphere InterChange Server or WebSphere Business Integration Server Express will be rendered as a standard JDBC resource in WebSphere Process Server, and the basic function will be the same. However, the way connections and transactions are managed may differ.

To maximize future portability, avoid keeping database transactions active across Java snippet nodes within a collaboration template or map. For example, code related to obtaining a connection, beginning and ending a transaction, and releasing the connection should be in one code snippet.

Premigration considerations: General development

Follow these recommended practices when developing WebSphere InterChange Server or WebSphere Business Integration Server Express modules to ease future migration to WebSphere Process Server.

Several considerations apply broadly to developing most of the integration artifacts. In general, artifacts that leverage the facilities provided by WebSphere InterChange Server or WebSphere Business Integration Server Express tooling and conform to the metadata models enforced by the tooling will migrate most smoothly. Also, artifacts with significant extensions and external dependencies are likely to require more manual intervention when migrating.

In general, IBM recommends that you do the following:

- Document the system and component design
- Use the development tooling to edit integration artifacts
- Leverage recommended practices for defining rules with the tooling and Java snippets

It is important for integration solutions to adhere to the programming model and architecture provided by WebSphere InterChange Server or WebSphere Business Integration Server Express. Each of the integration components within WebSphere InterChange Server or WebSphere Business Integration Server Express plays a well-defined role within the architecture. Significant deviations from this model will make it more challenging to migrate content to the appropriate artifacts on WebSphere Process Server.

Another general practice which will improve the success of future migration projects is to document the system design. Be sure to capture the integration architecture and design, including functional design and quality of service requirements, the interdependencies of artifacts shared across projects, and also the design decisions that were made during the deployment. This will assist in system analysis during migration, and will minimize any rework efforts.

For creating, configuring, and modifying artifact definitions, use only the development tooling provided. Avoid manual manipulation of artifact metadata (for example, editing XML files directly), which may corrupt the artifact for migration.

IBM suggests the following when you are developing Java code within collaboration templates, maps, common code utilities, and other components:

- Use only the published APIs.
- Use Activity Editor.
- Use adapters to access EISs.
- Avoid external dependencies in Java snippet code.
- Adhere to J2EE develop practices for portability.
- Do not spawn threads or use thread synchronization primitives. If you must, these will need to be converted to use Asynchronous Beans when you migrate.
- Do not do any disk I/O using java.io.* Use JDBC to store any data.
- Do not perform any functions that may be reserved for an EJB container such as socket I/O, classloading, loading native libraries, and so forth. If you must, these snippets would need manual conversion to use EJB container functions when you migrate.

Use only the APIs published in the WebSphere InterChange Server or WebSphere Business Integration Server Express product documentation for the artifacts. These are outlined in detail in the WebSphere InterChange Server or WebSphere Business Integration Server Express development guides. Compatibility APIs will be provided in WebSphere Process Server for published WebSphere InterChange Server or WebSphere Business Integration Server Express APIs. Although WebSphere InterChange Server and WebSphere Business Integration Server Express have many internal interfaces which you might wish to use, IBM discourages this practice because these interfaces are not guaranteed to be supported in the future.

When designing business logic and transformation rules in maps and collaboration templates, try to avoid field developed common code utility libraries, included as a Java archive (*.jar) file in the class path of WebSphere InterChange Server or WebSphere Business Integration Server Express, as these will need to be migrated manually.

Use the Activity Editor tool to the greatest extent possible. This will ensure that the logic is described through metadata which can more readily be converted to the new artifacts.

In any Java code snippets that may need to be developed, IBM recommends that the code be as simple and atomic as possible. The level of sophistication in the Java code should be on the order of scripting, involving basic evaluations, operations, and computations, data formatting, type conversions, and so forth. If more extensive or sophisticated application logic is required, consider using EJBs running in WebSphere Application Server to encapsulate the logic, and use web service calls to invoke it from WebSphere InterChange Server or WebSphere Business Integration Server Express. Use standard JDK libraries rather than third party or external libraries which would need to be migrated separately. Also, collect all related logic within a single code snippet, and avoid using logic where connection and transaction contexts span multiple code snippets. With database operations, for example, code related to obtaining a connection, beginning and ending a transaction, and releasing the connection should be in one code snippet.

In general, ensure that code which is designed to interface with an Enterprise Information System (EIS) is placed within adapters, and not within maps or collaboration templates. This is generally a recommended practice for architecture design. Also, this will help avoid prerequisites for third party libraries and related considerations within the code, such as connection management and possible Java Native Interface (JNI) implementations.

Make the code as safe as possible by using appropriate exception handling. Also make the code compatible to run within a J2EE application server environment, even though it is currently running within a J2SE environment. Adhere to J2EE development practices, such as avoiding static variables, spawning threads, and disk I/O. While these are generally good practices to adhere to, they can improve portability.

Premigration considerations: Maps

When developing WebSphere InterChange Server or WebSphere Business Integration Server Express maps, follow these guidelines to ensure the best chance of a smooth migration to WebSphere Process Server.

To ensure maps are described appropriately with metadata, always use the Map Designer or the Map Designer Express tool for the creation and modification of maps, and avoid editing the metadata files directly. Use the Activity Editor tool wherever possible to maximize the use of metadata to describe the required logic.

When referencing child business objects in a map, use a submap for the child business objects.

Avoid using Java code as the ″value″ in a SET since that is not valid in WebSphere Process Server. Use constants instead. For example, if the set value is ″xml version=″ + ″1.0″ + ″ encoding=″ + ″UTF-8″ this will not validate in WebSphere Process Server. Instead, change it to ″xml version=1.0 encoding=UTF-8″ before you migrate.

To minimize the amount of manual rework that may be required in migration, use only the documented APIs within maps. Avoid the use of static variables. Instead, use non-static variables. Avoid the use of Java qualifiers final, transient and native in map custom code.

If using an array in a business object, do not rely on the order of the array when indexing into the array in maps. The construct that this migrates into in WebSphere Process Server does not guarantee index order, particularly when entries are deleted.

To maximize future portability, avoid using explicit connection release calls and explicit transaction bracketing (that is, explicit commits and explicit rollbacks) for User Defined Database Connection Pools. Instead, make use of the container-managed implicit connection clean-up and implicit transaction bracketing. Also, avoid keeping system connections and transactions active in custom map steps across transformation node boundaries. This applies to any connection to an external system, as well as user-defined database connection pools. Operations with an external EIS should be managed within an adapter, and code related to database operation should be contained within one custom step.

Do not use inner classes in your maps. The migration command (reposMigrate) does not migrate inner classes and you will receive errors if your maps contain them. In a WebSphere InterChange Server or WebSphere Business Integration Server Express repository, an inner class could be defined in a node and referenced by other nodes within the same collaboration template. In WebSphere Process Server, an inner class defined in a BPEL component cannot be used by other components. Due to this limitation, inner classes are not translated and must be dealt with manually. Recommended changes include packaging the inner class code in a library as an external class, or removing the inner class declaration, resolving any errors, and placing the code as needed throughout the BPEL.

Premigration considerations: Preventing database collisions

Prevent database collisions from occurring by scheduling events to occur at least two seconds apart.

If your migrated applications cause multiple events to occur at the same time to WebSphere Business Integration components, this could cause database collisions, or deadlocks. These occur when the WebSphere Process Server Application Scheduler (AppScheduler) schedules multiple events to occur at exactly the same time. When a deadlock occurs, the event that caused it is rolled back and attempted again as soon as possible. This cycle continues until each of the threads attempting to access the database successfully updates it.

For example:

```
AppScheduler E com.ibm.wbiserver.scheduler.AppSchedulerMB process CWLWS0021E:
The AppSchedulerMB.process method has generated an exception.
WSRdbXaResour E DSRA0304E: XAException occurred. XAException contents and
  details are:
The DB2 Error message is : Error executing a XAResource.end(), Server returned
XA_RBDEADLOCK The DB2 Error code is : -4203
The DB2 SQLState is : null
```

To prevent this from occurring, schedule the events to occur far enough apart so that deadlocks do not occur. IBM recommends that you schedule events to occur at least two seconds apart; however, the amount of time you need will vary depending on other factors in your environment that affect performance such as database size, hardware, connection speed and other factors.

Premigration considerations: Relationships

While relationship definitions can be migrated for use in WebSphere Process Server, the relationship table schema and instance data may be reused by WebSphere Process Server, and shared concurrently between WebSphere InterChange Server or WebSphere Business Integration Server Express and WebSphere Process Server.

For relationships, use only the tooling provided to configure the related components, and use only the published APIs for relationships within integration artifacts.

Use only the Relationship Designer or the Relationship Designer Express tool to edit relationship definitions. In addition, allow only WebSphere InterChange Server or WebSphere Business Integration Server Express to configure the relationship schema, which is generated automatically upon deployment of relationship definitions. Do not alter the relationship table schema directly with database tools or SQL scripts.

If you must manually modify relationship instance data within the relationship table schema, be sure to use the facilities provided by the Relationship Manager.

Use only the published APIs for relationships within integration artifacts.

Migrating WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts with the reposMigrate command

Migrate WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts to WebSphere Process Server artifacts with the **reposMigrate** command.

Before you begin

Note: The functionality of the reposMigrate command is also available from WebSphere Integration Developer with a supporting wizard (graphical user interface). See the WebSphere Integration Developer information center for more information.

The **reposMigrate** command requires as input a WebSphere InterChange Server or WebSphere Business Integration Server Express repository JAR file. This JAR file should be self-contained with respect to the applications being migrated. That is, all artifacts referenced by any of the artifacts in the JAR file must also be contained in the JAR file.

To ensure that the repository JAR file that will be generated is self-contained, run the **repos_copy** command with the `-vr` option before exporting the server repository. This validates the repository. If the repository is valid then repos_copy writes the following output to the console: `Validation Succeeded. All Dependencies Resolved.` If the repository is not valid then **repos_copy** prints a list of the dependencies that must be resolved. Resolve the dependencies prior to exporting the repository.

Export the repository artifacts and create the repository JAR file, using the WebSphere InterChange Server or WebSphere Business Integration Server Express **repos_copy** command with the -o option (See the WebSphere InterChange Server or WebSphere Business Integration Server Express v4.3 documentation for more details, including how to export individual components).

About this task

The **reposMigrate** command will convert the WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts in a JAR file into WebSphere Process Server deployable artifacts. These artifacts are modules created as one or more JAR files. A JAR file is created for each collaboration object and for each connector definition that has been migrated. For other artifacts such as business objects, maps and relationships, a copy of all of these artifacts generated from the input JAR file will be included in each JAR file generated. If no collaboration objects or connectors are migrated, a single JAR file is created containing a module of all the shared artifacts. After the new JAR files are created, you will use the **serviceDeploy** command to generate the EAR files that can be deployed in WebSphere Process Server.

For WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts that have no corresponding artifact in WebSphere Process Server, a Jython script is generated during migration that can be run using the **wsadmin** command to create WebSphere Process Server configuration definitions corresponding to the original WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts.

Procedure

1. Identify the JAR file containing the pre-exported WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts that are to be converted to WebSphere Process Server deployable artifacts.
2. Invoke the **reposMigrate** command from a command-line prompt. Type the command at a command prompt in WebSphere Process Server, with the required arguments and any optional arguments you require. Refer to reposMigrate command for more information.
3. If desired, edit the resulting JAR file.
4. Run serviceDeploy to create a deployable EAR file for each JAR file.

   Note: The support in the WebSphere Process Server Runtime to handle migrated WebSphere InterChange Server applications relies on the default naming convention used by the serviceDeploy command. IBM recommends that you do not specify the **serviceDeploy -outputApplication** parameter when building migrated projects with the **serviceDeploy** command, so that it will generate its default output filenames.

   For more information, refer to the WebSphere Process Server serviceDeploy command in the *Reference* PDF file.
5. Use the administrative console or the **wsadmin** command to install the EAR files on WebSphere Process Server. Use the **wsadmin** command to execute the InstallAdministrativeObjects.py script. This will create resources in the WebSphere Process Server system for all target resources such as JDBC data sources and WBIScheduler entries.

Example

You can use the **reposMigrate** command to migrate existing WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts directly to a running WebSphere Process Server:

1. Open a command prompt in WebSphere Process Server.
2. Issue the **reposMigrate** command with the following mandatory parameters:

   install_root\bin\reposMigrate *SourceArtifactJAR OutputArtifactDirectory*

The reposMigrate command builds the generated artifacts as follows:

- For each WebSphere InterChange Server or WebSphere Business Integration Server Express collaboration object and connector definition in the input JAR file, **reposMigrate** creates a JAR file from the migrated artifacts.
- For other artifacts such as business objects, maps and relationships, a copy of all of these artifacts generated from the input JAR file will be included in each JAR file generated. If no collaboration objects nor connector definitions were in the input, a single JAR file will be created with all the shared artifacts.

What to do next

The default behavior of the **reposMigrate** command is to log errors for the migration of each individual artifact and continue to migrate the remainder of the artifacts. You should check output messages for errors after the execution completes. To override this default behavior and force **reposMigrate** to end processing when the first artifact that cannot be migrated is encountered, specify the **-fh** (halt at first failure) flag. You can run **reposMigrate** from the beginning to retry after a failed execution.

Related reference

Postmigration considerations
When applications have been migrated from WebSphere InterChange Server or WebSphere Business Integration Server Express to WebSphere Process Server, special attention is required in some areas to enable migrated applications to function in WebSphere Process Server consistently with their intended function due to differences between the architectures of WebSphere Process Server and WebSphere InterChange Server or WebSphere Business Integration Server Express.

Related information

Wsadmin tool

reposMigrate command
WebSphere InterChange Server v4.3 documentation

WebSphere Integration Developer information center

Postmigration considerations

When applications have been migrated from WebSphere InterChange Server or WebSphere Business Integration Server Express to WebSphere Process Server, special attention is required in some areas to enable migrated applications to function in WebSphere Process Server consistently with their intended function due to differences between the architectures of WebSphere Process Server and WebSphere InterChange Server or WebSphere Business Integration Server Express.

You should be aware of the information described in the following sections if it applies to your application and environment:

"Security" on page 158

"Handling existing database connections, relationships, and scheduled events (InstallAdministrativeObjects.py script)" on page 158

"Handling existing WebSphere InterChange Server or WebSphere Business Integration Server Express database connection pools" on page 159

"Using an existing WebSphere InterChange Server or WebSphere Business Integration Server Express relationship database" on page 159

"Migrating scheduled events" on page 160

"Access Enterprise JavaBean (EJB) support" on page 160

"DynamicSend API configuration" on page 161

"Enabling the BaseCollaboration.dynamicSend method call" on page 161

"Event sequencing migration" on page 163

"Failed events" on page 164

"Map migration" on page 164

"Collaboration migration" on page 164

"BPEL variables must be defined after migration" on page 165

"Enabling logError API e-mail notification on WebSphere Process Server" on page 165

"Handling asynchronous calls in WebSphere Process Server" on page 166

"Enabling AppScheduler to start after network deployment upgrade" on page 166

"Handling correlation values in WebSphere Process Server" on page 167

"Packaging and deploying migrated applications" on page 167

Security

Additional security configuration is required for your applications to have the same security levels set as they had when running in WebSphere InterChange Server or WebSphere Business Integration Server Express. For details on this configuration, refer to "Configuring global security after WebSphere InterChange Server or WebSphere Business Integration Server Express migration" on page 168.

Handling existing database connections, relationships, and scheduled events (InstallAdministrativeObjects.py script)

The Jython script InstallAdministrativeObjects.py is generated during migration. This script has three purposes: It allows migration of WebSphere InterChange Server or WebSphere Business Integration Server Express scheduler entries that have no corresponding artifact in WebSphere Process Server; it allows the use of existing DBConnection pools; and it allows the use of an existing relationship database. You can run the script with the wsadmin command to create WebSphere Process Server configuration definitions corresponding to the original WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts. A copy of InstallAdministrativeObjects.py is included wherever the shared artifacts are included. That is, the script is included in every JAR file created by the reposMigrate command, and it is put in the shared library project specified during import in WebSphere Integration Developer. An InstallAdministrativeObjects.py script is always generated even if there are no artifacts that require it. This script can be modified to add or delete entries before using the wsadmin command to execute it.

For more information about using the wsadmin command, see wsadmin tool .

Handling existing WebSphere InterChange Server or WebSphere Business Integration Server Express database connection pools

To preserve existing WebSphere InterChange Server or WebSphere Business Integration Server Express DataBase Connection pools for use by WebSphere Process Server you can run the InstallAdministrativeObjects.py script with the wsadmin command to create the connection pools in WebSphere Process Server. If an appropriate JDBC provider is not defined, this script will use default JDBC provider templates to create JDBC providers. A side effect of using these default templates is that WebSphere Process Server creates an empty, sample data source definition. This sample data source is not used; you must delete it to prevent exceptions from occurring during server start because it does not specify all of the information required for a data source.

In the WebSphere InterChange Server or WebSphere Business Integration Server Express environment, resources are defined only once for the entire system. To simulate this in the WebSphere Process Server environment, the InstallAdministrativeObjects.py script defines resources at the cell scope. WebSphere variables are predefined at the node scope in the WebSphere Process Server system for use by JDBC providers created from the default JDBC provider templates. These variables are defined at the node scope so that they can be customized for each node. Because of this scoping discrepancy, you will need to do one of the following:

- Define the WebSphere Variables needed by the created JDBC providers at the cell scope.
- Run the InstallAdministrativeObjects.py script and then move the JDBC providers to the node scope.

Use the administrative console to examine the JDBC providers that are generated to determine which WebSphere variables are needed. From the administrative console, select **Environment > WebSphere Variables** to create any required variables. For more information, see Defining WebSphere variables in the WebSphere Application Server Network Deployment, version 6.1 information center.

Here is an example of what the generated InstallAdministrativeObjects.py script might contain to generate the JDBC connector pool:

```
dsName = "sqls"
create_datasource(dsName, JNDI_PREFIX + dsName, DATASOURCE_DESCRIPTION,
MS_SQL_JDBC_PROVIDER_NAME, MS_SQL_JDBC_PROVIDER_TYPE, "icsadmin",  "icsadmin",
4, 50, "qaxs17", "1433", "wicsrepos")
```

For more information about the wsadmin command, see wsadmin tool .

Using an existing WebSphere InterChange Server or WebSphere Business Integration Server Express relationship database

To use an existing WebSphere InterChange Server or WebSphere Business Integration Server Express relationship database in WebSphere Process Server, you can use the InstallAdministrativeObjects.py script with the wsadmin command to create the data source and relationship configuration information in WebSphere Process Server. Normally, WebSphere Process Server automatically creates the configuration information for the migrated relationships when they are deployed. To be able to use the existing database, the InstallAdministrativeObjects.py script has to create the database connection for the existing WebSphere InterChange Server or WebSphere Business Integration Server Express relationship database and the relationship configuration information in WebSphere Process Server. Runs the InstallAdministrativeObjects.py script before you deploy the migrated components. Then, when WebSphere Process Server deploys the relationships it will use the configuration information that was generated by the script.

Here is an example of what the generated InstallAdministrativeObjects.py script might contain to generate the relationship database connection:

```
dsName = "ContactR"
create_datasource(dsName, JNDI_PREFIX + dsName, DATASOURCE_DESCRIPTION,
MS_SQL_JDBC_PROVIDER_NAME, MS_SQL_JDBC_PROVIDER_TYPE, "icsadmin",  "icsadmin",
-1, -1, "9.26.230.56", "1433", "wicsrepos")

create_relationship("ContactR", "jdbc/wbi60migration/ContactR", "false")
create_role("ContactR", "ID1", "", "null", "", "null")
create_attribute("ContactR", "ID1", "JtextEmployeeID")
create_role("ContactR", "ID2", "", "null", "", "null")
create_attribute("ContactR", "ID2", "EmployeeID")
create_role("ContactR", "ID3", "", "null", "", "null")
create_attribute("ContactR", "ID3", "EmployeeID")
```

For more information about the wsadmin command, see wsadmin tool .

Migrating scheduled events

Because there is no WebSphere Process Server component that corresponds to WebSphere InterChange Server or WebSphere Business Integration Server Express scheduler entries, migration of WebSphere InterChange Server or WebSphere Business Integration Server Express scheduler entries is accomplished by extracting the pertinent data from the existing WebSphere InterChange Server or WebSphere Business Integration Server Express repository JAR file and creating corresponding entries in the WebSphere Process Server scheduler tables in the WebSphere Process Server Common database. The data is represented in string form in the Jython script. To create the scheduler entries in the WebSphere Process Server database, you can run the InstallAdministrativeObjects.py script with the wsadmin command.

Here is an example of what the generated InstallAdministrativeObjects.py script might contain to generate the scheduler entry:

```
create_scheduler_entry("true", "stop", "JDBCConnector", "Connector",
"2006-09-07T10:44:29.000PDT", "undefined", 0, 0)
   create_scheduler_entry("true", "start", "JTextConnector", "Connector",
"2006-09-07T10:47:06.000PDT", "undefined", 0, 0)
   create_scheduler_entry("true", "stop", "jtext_jdbcCollab", "Collaboration",
"2006-09-07T10:48:10.000PDT", "undefined", 0, 0)
   create_scheduler_entry("true", "start", "jtext_jdbcCollab", "Collaboration",
"2006-09-07T10:48:10.000PDT", "undefined", 0, 0)
   create_scheduler_entry(true, "START", "JDBCConnector", "Connector",
"2006-10-22T12:34.56.789CDT", "MINUTES", 20, 0):
```

Access Enterprise JavaBean (EJB) support

WebSphere InterChange Server or WebSphere Business Integration Server Express supports the triggering of collaborations by client code via the J2EE EJB (Enterprise

JavaBeans™) protocol. Support for this method of triggering collaborations is referred to as ″AccessEJB″ or ″AccessEJB for EJB″ support. For compatibility with earlier versions, WebSphere Process Server provides support for AccessEJB. The AccessEJB support assumes that the SCA BPEL modules to be invoked were generated by the WebSphere InterChange Server or WebSphere Business Integration Server Express migration tools described in this documentation. The mapping from the collaboration name and port name (that is, the input parameters for the AccessEJB) to the SCA module name, interfaces and business object types assume the conventions used by the migration tools. The AccessEJB support in WebSphere Process Server is delivered in the AccessEJB.zip project interchange file. This file is located in the *install_root*/HeritageAPI directory. The AccessEJB support consists of an EJB (AccessEJB) which references an SCA module project (DynamicRouting) that invokes the SCA BPEL module. This SCA BPEL module is the migrated version of the collaboration that was invoked in WebSphere InterChange Server or WebSphere Business Integration Server Express. The DynamicRouting module uses a selector component to select the correct SCA target based on the collaboration name and port name passed to the AccessEJB. To enable AccessEJB support in WebSphere Process Server, do the following:

1. Import the WebSphere InterChange Server or WebSphere Business Integration Server Express repository containing the collaboration that is the target of the AccessEJB invocation into WebSphere Integration Developer.
2. Import the AccessEJB.zip project interchange file into WebSphere Integration Developer.
3. Open the DynamicRouting project and update the selector table to include the migrated module that is to be invoked via the AccessEJB.
4. Go to the migrated project containing the BPEL component to be invoked via the AccessEJB EJB, and drag the export that references the BPEL module over to the DynamicRouting project.
5. Repeat steps 3 and 4 for each BPEL module that is to be accessible via AccessEJB.
6. Build the project and deploy it to the WebSphere Process Server server.
7. Ensure that any required data handlers are provided in the runtime class path of the WebSphere Process Server server.
8. To enable your Access client to use WebSphere Process Server, ensure that it points to the WebSphere Process Server server and uses the JNDI name com/crossworlds/access/business/cwsession/CwSession when looking up the Access EJB.

DynamicSend API configuration

In WebSphere InterChange Server or WebSphere Business Integration Server Express, the DynamicSend API can be used to directly invoke one collaboration from another. The collaboration to be invoked does not have to be predetermined; instead, it can be determined dynamically at runtime. The support for the DynamicSend API in WebSphere Process Server uses the DynamicRouting project described in ″“Access Enterprise JavaBean (EJB) support” on page 160.″ Follow the instructions in “Enabling the BaseCollaboration.dynamicSend method call” to enable the DynamicSend API to be able to invoke the specified BPEL modules.

Enabling the BaseCollaboration.dynamicSend method call

To enable the WebSphere InterChange Server or WebSphere Business Integration Server Express BaseCollaboration.dynamicSend method call to work correctly after migration you must modify the DynamicRouting Projects in the AccessEJB Project Interchange file. This requires two main procedures:

1. Migrating the WebSphere InterChange Server or WebSphere Business Integration Server Express repository.
2. Enabling the DynamicSend API.

To migrate the WebSphere InterChange Server or WebSphere Business Integration Server Express repository:

1. Import the WebSphere InterChange Server or WebSphere Business Integration Server Express repository containing the collaboration that invokes DynamicSend API into WebSphere Integration Developer.
2. Import the WebSphere InterChange Server or WebSphere Business Integration Server Express repository containing the collaboration or connector that is the target of the DynamicSend API invocation into WebSphere Integration Developer.
3. Build all, and correct all errors.

To enable the DynamicSend API:

1. Import the AccessEJB.zip project interchange file into WebSphere Integration Developer.
2. Open the DynamicRouting project, and add the WebSphere InterChange Server or WebSphere Business Integration Server Express shared library into the DynamicRouting project's dependencies.
3. Go to the migrated module containing the component to be invoked through the BaseCollaboration.dynamicSend method, and drag the export that references the module over to the DynamicRouting project. Choose **Import with SCA Binding** and then click **OK**.
4. In the DynamicRouting Assembly Diagram window, copy and paste `PreRoute_TargetCollab_TargetPort` and then rename the newly created copy to `PreRoute_`*`ModuleName_ExportName`* (the name of the copied import will be `PreRoute_TargetCollab_TargetPortCopy`).
5. On `PreRoute_`*`ModuleName_ExportName`*, left click on the reference, which is the small box attached to the right containing `1.1`. Right click and choose **Delete**.
6. Wire the `PreRoute_`*`ModuleName_ExportName`* to the import generated in step 3. Respond with "no" to the Java WSDL reference question.
7. Rename the import to *`ModuleName_ExportName`*. Save the changes to the Assembly Diagram.
8. Update the selector table in the DynamicRouting project to include the migrated module that is to be invoked through the DynamicSend API.
   a. Switch to the Java Perspective Package Explorer View. Expand DynamicRouting/com.ibm and open RoutingSelector.selt with the text editor.
   b. Copy the OperationSelectionRecord block and paste the entire block immediately following the existing block.
   c. In the new block, change `componentName="PreRoute_TargetCollab_TargetPort"` to `componentName="PreRoute_`*`ModuleName_ExportName`*`"`. Also in the new block, change `value="TargetCollab_TargetPort"` to `value="`*`ModuleName_ExportName`*`"`.

```
<OperationSelectionRecord>
      <SelectionKey>
      <SelectionKeyElement xsi:type="selt:StringSingletonKey" value=
           "TargetCollab_TargetPort"/>
      </SelectionKey>
      <SelectionData xsi:type="selt:SCAInternalComponent"
            componentName="PreRoute_TargetCollab_TargetPort"/>
      </OperationSelectionRecord>
      <OperationSelectionRecord>
      <SelectionKey>
      <SelectionKeyElement xsi:type="selt:StringSingletonKey"
             value="ModuleName_ExportName"/>
      </SelectionKey>
      <SelectionData xsi:type="selt:SCAInternalComponent" componentName
             ="PreRoute_ModuleName_ExportName"/>
</OperationSelectionRecord>
```

   d. Save and close RoutingSelector.selt.

9. Generate the implementation file.
   a. Expand **com.ibm.sel** and copy PreRoute_TargetCollab_TargetPortImpl.java and paste it to the same location. Name the newly created Java file PreRoute_*ModuleName_ExportName*Impl.java.
   b. Edit PreRoute_*ModuleName_ExportName*Impl.java. Change the method name of `locateService.TestBOInterfacePartner` to `locateService_`*`InterfaceName`*`Partner` (*InterfaceName* is the method). Change `TestBOInterfacePartner` to *`InterfaceName`*`Partner`.
   c. Search for ″`locateService_TestBOInterfacePartner`″ in PreRoute_*ModuleName_ExportName*Impl.java, and change its name to `locateService_`*`InterfaceName`*`Partner`.
10. Switch back to the Business Integration Perspective. Open the DynamicRouting Assembly Diagram. Click on **PreRoute_*ModuleName_ExportName***. Open **Properties** and select **Implementation**. In the **Class:** field, enter `com.ibm.sel.PreRoute_`*`ModuleName_ExportName`*`Impl`.
11. Save all changes.
12. Repeat steps 3 to 11 for any other modules that you want to call from the BaseCollaboration.dynamicSend Method. There is currently no way to ″dynamically look these modules up″ if you do not add them to the DynamicRouting Table so you can access them during run time.
13. For the project that calls the dynamicSend API, do the following
   a. Copy and paste Interface ″RoutingPacket″ from Module DynamicRouting.
   b. In the component that calls the dynamicSend method. Add the newly copied interface ″RoutingPacket″ to Reference_Partners, and rename it ″RoutingPacketPartner.″
   c. Save it.
   d. Open the Assembly Diagram. Drag ″RoutingInput″ from DynamicRouting. Choose ″Import with SCA Binding″ and click ″OK″. Rename it from ″Import1″ to ″DynamicRouting″.
   e. Delete and re-drag the component that calls the dynamicSend API to the Assembly Diagram window, wire the Reference ″RoutingPacketPartner″ to ″DynamicRouting″ and re-wire the other references.
14. Save all and build, then correct all errors. Export all the modules to EAR files.

Event sequencing migration

Methods are available for sequencing events with WebSphere Process Server in ways similar to the way you could with WebSphere InterChange Server or WebSphere Business Integration Server Express. Articles on this subject that you might find helpful are available from the IBM developerWorks web site. Search in the ″Technical Library″ at http://www.ibm.com/developerworks.

Failed events

Methods for handling failed events in WebSphere Process Server are described in article(s) that you might find helpful on the IBM developerWorks web site. Search in the ″Technical Library″ at http://www.ibm.com/developerworks.

Map migration

WebSphere InterChange Server or WebSphere Business Integration Server Express migration converts WebSphere InterChange Server or WebSphere Business Integration Server Express maps into WebSphere Process Server maps. Two output maps are generated: the business graph map and the business object map. The business graph map calls the business object map as a submap. All the business graph maps are identical in structure. Differences include names, names of the submap they call, or ASI information on the verb attribute. These business graph maps are present only to satisfy the necessary mapping steps that can only be done at the business graph level. The business object maps are each unique and are the migrated form of the WebSphere InterChange Server or WebSphere Business Integration Server Express map. If the WebSphere InterChange Server or WebSphere Business Integration Server Express input map contains custom messages for the supported WebSphere InterChange Server or WebSphere Business Integration Server Express API log methods, these messages will be converted into a properties file.

Collaboration migration

Collaboration Templates: The WebSphere InterChange Server or WebSphere Business Integration Server Express to WebSphere Process Server migration tools migrate WebSphere InterChange Server or WebSphere Business Integration Server Express Collaboration templates into WebSphere Process Server BPEL files. One BPEL file is created for each triggering port defined in a collaboration template, and its name is based on the following naming convention: *CollaborationTemplateName_TriggeringPortName*. Each BPEL file receives a business object type that is based on the business object type associated with the triggering port. For example, if the triggering port takes a business object type of Customer, then the BPEL file that is created will have a “TriggeringBusObj” variable type of Customer.

Collaboration Objects: The WebSphere InterChange Server or WebSphere Business Integration Server Express to WebSphere Process Server migration tools migrate collaboration objects into several service component architecture (SCA) components. Currently, migration supports collaboration objects that reference collaboration templates as follows:

- Supported:
  - One or more triggering ports, no correlation sets and no asynchronous in calls
  - Exactly one triggering port, correlation sets and asynchronous in calls
- Unsupported:
  - Migration does not support the case of one or more triggering ports, correlation sets and asynchronous in calls. In this case, the resulting artifacts are migrated as if they were the first case listed above. Additionally you will need to manually create the missing SCA components and wire them together appropriately.

SCA Components:

- Exports: An export is created for every triggering port defined in the collaboration template associated with the collaboration object. The export name is *TriggeringPortName*.
- Export to BPEL: An interface map is generated that maps the data from the export to the BPEL file. The interface map name is *Export_To_BPELname.* When there is exactly one triggering port and the collaboration template has an asynchronous in call, additional SCA components are created. Instead of having just one interface map, the migration results in two interface maps: one for synchronous calls, and another for asynchronous calls. A Java component is used to decide which of these two interface maps to follow.
- BPEL: For every triggering port, the export will be wired to an interface map and that interface map will map to an instance of the BPEL file.
- BPEL to import: Every port, triggering and non-triggering, has an interface map mapping the BPEL file to the import. The interface map name is *BPEL_to_Port*.
- Import: Finally, an import file is created. The import name is *ConnectorName_BONameBG*.

For further detailed information on how collaboration templates are migrated to WebSphere Process Server BPEL files, see the IBM developerWorks article Migrating WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts to WebSphere Process Server artifacts, Part 1: Migrating collaboration templates to BPEL .

BPEL variables must be defined after migration

Problem: A variable that is not defined in the Ports definitions of the WebSphere InterChange Server or WebSphere Business Integration Server Express Collaboration Template is used to invoke a partner. After migration, the variable is referenced in the business process execution language (BPEL) invoke but has not been set up as a BPEL variable, so it is flagged as an error when executing the serviceDeploy command against the module or after building the module in WebSphere Integration Developer. **Cause:** When invoking a partner from a BPEL process in WebSphere Process Server, any object used in the invoke must be declared as a BPEL variable so that the type of the object being used can be determined. During migration, only the Ports declarations in the Collaboration Template are examined to determine what BPEL variables need to be declared. For global variables, or variables declared in snippets elsewhere in the ICS Collaboration Template definition, the migration code cannot reliably determine the object type, so BPEL variables are not declared for these in the BPEL file generated by migration. **Solution:** After migration, you must define the variable as a BPEL variable for the variable to be referenced during an invoke.

Enabling logError API e-mail notification on WebSphere Process Server

Problem: After migration to WebSphere Process Server, the WebSphere InterChange Server or WebSphere Business Integration Server Express logError API does not send an e-mail to a list of users that has been configured in WebSphere InterChange Server or WebSphere Business Integration Server Express. **Cause:** In WebSphere InterChange Server or WebSphere Business Integration Server Express, you could configure the API call logError to send an error e-mail to a specified list of users. However, this list of users, configured on the server, is not accessible to the migration code so it must be set up manually in WebSphere Process Server. **Solution:**To enable the WebSphere InterChange Server or WebSphere Business

Integration Server Express logError e-mail notification functionality in WebSphere Process Server, a new BPEL environment variable called *LOGERROR_EMAIL_LIST* is created in each BPEL file generated by migration. Set this variable with the list of e-mail users needing to receive log error e-mails. Separate the names within the list with a comma.

Handling asynchronous calls in WebSphere Process Server

Problem: Async-in events act as triggering events when both types of events can be received on the same connector. **Cause:** If both async-in and triggering events can be received on the same connector, the migrated application cannot determine which events are which type. By default, all events are treated as triggering events in a migrated application in this scenario. **Solution:** Application-specific logic that can determine if an event is async-in or triggering must be added to the migrated application. Migrated modules that can receive triggering events and async-in events on the same connector will have a component named JavaSelector. The implementation code for the JavaSelector component will contain the AsyncIn() method shown below. This method must be updated with logic to check if events are async-in or triggering. This logic will be specific to each application and will be based on the nature of the events being handled.

```
/** * Method generated to support async inbound service call routing */
public boolean isAsyncIn()
{ //Add custom code here
   //TODO
   return false;
}
```

Enabling AppScheduler to start after network deployment upgrade

Problem: After migrating a WebSphere Process Server 6.0.1.x network deployment configuration to WebSphere Process Server 6.1, the AppScheduler fails to start on WebSphere Process Server 6.0.1.x servers and clusters that have not been upgraded. An Exception similar to the following will be generated:

```
WSVR0040E: addEjbModule failed for WBISchedulerEJB.jar
   [class com.ibm.ws.runtime.component.
DeployedEJBModuleImpl] java.lang.NoClassDefFoundError:
   com/ibm/wbiserver/scheduler/common/AppSchedulerException
```

Cause: After migrating the WebSphere Process Server 6.0.1.x network deployment configuration to WebSphere Process Server 6.1, the AppScheduler application looks for the AppSchedulerException Class in the WebSphere Process Server 6.0.1.x version of the wbischedulercommon.jar file and fails to find it in the local system's *install_root*/lib directory. It then throws a java.lang.NoClassDefFoundError: com/ibm/wbiserver/scheduler/common/AppSchedulerException exception. **Solution:** Replace the WebSphere Process Server 6.0.1.x version of the wbischedulercommon.jar file with the WebSphere Process Server 6.1 or WebSphere Process Server 6.0.2.x version of that JARfile. You can obtain the new JAR file from the WebSphere Process Server 6.1.x *install_root*AppScheduler/lib directory or the WebSphere Process Server 6.0.2 *install_root*/lib directory. Copy the JAR file into the WebSphere Process Server 6.0.1.x lib directory and replace the existing JAR file. Do not rename the existing JAR file and leave it in the lib directory, because WebSphere Process Server picks up all files in the lib directory as JAR files regardless of the extension. Then, restart the server or cluster so that WebSphere Process Server picks up the new JAR file.

Handling correlation values in WebSphere Process Server

Problem: In WebSphere Process Server, new events attempting to use the existing correlation values will fail. In such instances, the error message

```
CWWBE0074E: Correlation violation in activity 'null' for correlation set
'CorrelationSetA'java.sql.
SQLException: Could not insert new row - duplicate value in a UNIQUE INDEX column
```

appears. **Cause:** When a collaboration or process instance completes in WebSphere InterChange Server or WebSphere Business Integration Server Express, data related to that instance is deleted except for cases dealing with failures. In WebSphere Process Server, the persistence of process instance-related data is controlled by the business process execution language (BPEL) option, ″Automatically delete the process after completion.″ BPEL files generated by the WebSphere InterChange Server or WebSphere Business Integration Server Express to WebSphere Process Server migration wizard will not have this option selected. As a result, process instance data will persist, even after the process instance completes, until you clean it up manually. When a process defines a correlation set, the correlation values locked by process instances remain locked as long as the process instance data is persisted, even after the process has completed. As a result, new events attempting to use the same correlation values will fail as long as the previous process instance's data persists. This behavior will be different than in WebSphere InterChange Server or WebSphere Business Integration Server Express, where new events with duplicate correlation set values could be processed as soon as the previous instance was complete. **Solution:** To simulate the behavior of WebSphere InterChange Server or WebSphere Business Integration Server Express with respect to multiple events with duplicate correlation set values, you can choose to select the BPEL option ″Automatically delete the process after completion″ so that process instance data is deleted, and the correlation value is unlocked, as soon as the process instance completes. Before selecting this option, you should investigate and fully understand the way failures are handled in WebSphere Process Server and to ensure that your failed event strategy does not rely on data that will be automatically deleted when this option is set.

Packaging and deploying migrated applications

After migrating the WebSphere InterChange Server or WebSphere Business Integration Server Express repository using the reposMigrate command, you will need to package the resulting JAR files into EAR files in order for them to be deployed to the WebSphere Process Server. To do this, you can either import each migration-generated JAR file into WebSphere Integration Developer and export the modules as EAR files, or you can use the serviceDeploy command. The serviceDeploy command accepts a JAR file as input and outputs a deployable EAR file. Packaging the migration code into EAR files involves compiling the resulting migrated JAR file. If this produces validation errors, they are most likely due to the use of unsupported WebSphere InterChange Server or WebSphere Business Integration Server Express APIs or third-party APIs that were present in WebSphere InterChange Server or WebSphere Business Integration Server Express but have not yet been included in the WebSphere Process Server class path. Remove the unsupported APIs and add the third-party classes to the WebSphere Process Server class path.

Validation errors may also be caused by not following premigration recommended practices or may indicate postmigration work that still needs to be performed on the artifacts. As with migration errors, each validation error should be handled on a per-error basis. If a recommended premigration practice was not followed, you can update the repository and migrate it again, or you can edit the output artifacts to remove the problem.

Any other validation errors should be resolved as if these artifacts were created from scratch. You should refer to the validator documentation that outlines common artifact errors and their resolutions. Inevitably an automated migration cannot account completely for your program's intent; it can just make best guesses. Therefore, even if there are no validation errors, it is possible the migrated artifacts do not perform as intended. You should review all artifacts to confirm that the intended purpose of each artifact is met by its migrated content.

Related concepts

Limitations when migrating from WebSphere InterChange Server or WebSphere Business Integration Server Express
Some characteristics of WebSphere InterChange Server or WebSphere Business Integration Server Express are not precisely duplicated by WebSphere Process Server. Therefore you might have to modify your applications after migration to get them to perform as they did in WebSphere InterChange Server or WebSphere Business Integration Server Express.

Troubleshooting migration from WebSphere InterChange Server or WebSphere Business Integration Server Express
Find solutions to problems you encounter with migration as well as instructions for turning on logging and tracing.

Related tasks

Migrating WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts with the reposMigrate command
Migrate WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts to WebSphere Process Server artifacts with the **reposMigrate** command.

Related reference

Premigration considerations
Consider these guidelines for the development of integration artifacts for WebSphere InterChange Server or WebSphere Business Integration Server Express to ease the migration of WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts to WebSphere Process Server.

Related information

serviceDeploy command

Wsadmin tool

WebSphere Integration Developer information center

IBM developerWorks

Migrating WebSphere InterChange Server artifacts to WebSphere Process Server artifacts, Part 1: Migrating collaboration templates to BPEL

Defining WebSphere variables

Configuring global security after WebSphere InterChange Server or WebSphere Business Integration Server Express migration

Perform these additional security configuration steps to enable projects migrated from WebSphere InterChange Server or WebSphere Business Integration Server Express to run successfully in a WebSphere Process Server environment.

Before you begin

You must first configure security for your WebSphere Process Server as described in Securing applications and their environment. In particular, make sure you have completed the steps in Securing adapters and Creating end-to-end security. In addition, install the EAR file for each module. Refer to Deploying (installing) secure applications for details.

About this task

After performing the above tasks, you are ready to complete the configuration steps, as follows:

- Binding the message-driven bean to activation specification
- Mapping the resource references to resources
- Mapping security roles to users or groups (required only when monitoring Common Based Events)
- Mapping RunAs roles (required only when monitoring Common Based Events)

Note: Mapping security roles to users or groups and mapping RunAs roles is possible from the administrative console only if the EJB deployment descriptors for the EJB projects have had a RunAs role defined. See Mapping users to RunAs roles using an assembly tool in the WebSphere Application Server Network Deployment, version 6.1 information center for information about defining RunAs roles with an assembly tool.

Procedure

1. Bind the message-driven bean to activation specification.
   a. From the administrative console, select **Applications > Enterprise Applications**.
   b. In the right panel, select the name of the application you just installed. (Select the name, not the check box to the left of the name.)
   c. In the right panel again, under Enterprise Java Bean Properties, select **Message Driven Bean listener bindings**.
   d. For each import or export EJB (indicated by an EJB name that starts with ″_import″ or ″_export″), under the Bindings column, specify **SCA_Auth_Alias** in the ″ActivationSpec authentication alias″ field.
   e. Select **OK**, then **Save**.
2. Map the resource references to resources.
   a. From the administrative console, select **Applications > Enterprise Applications**.
   b. In the right panel, select the name of the application you just installed. (Select the name, not the check box to the left of the name.)
   c. In the right panel, under References, select **Resource references**.
   d. In the Specify authentication method: field under javax.jms.ConnectionFactory, select the **Use default method (many to one mapping)** radio button.
   e. In the **Select authentication data entry** pull down menu, select **SCA_Auth_Alias**.
   f. Check the check box to select all of the modules.
   g. Select **Apply**, then **OK**, then **Save**.
3. Map security roles to user groups.

   a. From the administrative console, select **Applications > Enterprise Applications**.
   b. In the right panel, select the name of the application you just installed. (Select the name, not the check box to the left of the name.)
   c. In the right panel, under Detail Properties, select **Security role to user/group mapping**.
   d. Select the check box to the left of the role you wish to map and then select **Look up users**.
   e. Select **Search** to display a list of users who are available to map to the role, and move the correct user name to the ″Selected:″ column.
   f. Select **OK**. The ″Security role to user/group mapping″ panel will be redisplayed.
   g. Uncheck the check boxes in the ″Everyone?″ and ″All authenticated?″ columns corresponding to the role, and select **OK**, then **Save**.
4. Map RunAs roles.
   a. From the administrative console, select **Applications > Enterprise Applications**.
   b. In the right panel, select the name of the application you just installed. (Select the name, not the check box to the left of the name.)
   c. In the right panel, under Detail Properties, select **User RunAs roles**.
   d. Select the check box next to the role you mapped in step 3 on page 169.
   e. Enter the user name and password corresponding to the user name selected in step 3e into the username and password fields, respectively.
   f. Select **Apply**.
   g. Select **OK**, then **Save**.

What to do next

After installing and configuring all of the EAR projects, select **Applications > Enterprise Applications** in the administrative console and start the installed migrated projects. If they start successfully, then you are now ready to send events through one of the inbound connectors to be processed by the server.

Related information

Wsadmin tool

WebSphere InterChange Server v4.3 documentation

Mapping users to RunAs roles using an assembly tool

Securing adapters

Securing applications and their environment

Deploying (installing) secure applications

Creating end to end security

Support for WebSphere Business Integration data handlers

The data handler support API enables certain data handler methods to be invoked from the AccessEJB, a WebSphere Process Server SCA Java component, or WebSphere Process Server bindings.

WebSphere Process Server (version 6.0.2.3 and higher) provides a data handler support Application Programming Interface (API) which enables select WebSphere Business Integration data handler methods to be invoked from the AccessEJB, a WebSphere Process Server SCA Java component, or WebSphere Process Server bindings. Access EJB has been replicated as an EJB that allows JService calls to route the input business object to the appropriate migrated module. The BPEL file in the migrated module will be invoked instead of the original WebSphere InterChange Server or WebSphere Business Integration Server Express target collaboration.

WebSphere Process Server bindings invoke data bindings to perform data transformation.WebSphere Process Server provides several built-in data bindings as well as the capability to provide user-defined data bindings. You can implement a user-defined, or custom data binding to invoke a WebSphere Business Integration data handler.

By providing a custom data binding implementation, it is possible to leverage WebSphere Business Integration data handlers via the data handler support API. The data handler support API provides wrapper methods around existing WebSphere Business Integration data handler interface methods which perform the conversion between WebSphere Business Integration business objects and SDOs.

Data handler support API

By providing a custom data binding implementation, it is possible to leverage WebSphere Business Integration data handlers via the data handler support API. This API defines a set of public methods which can be invoked from a custom data binding or a Java component. It provides a way to invoke a text-based WebSphere Business Integration data handler from a process server binding. The following are the API methods:

`getSDOFromString(String` *`inputString`*`, String` *`sdoName`*`, String` *`metaObjectName`*`, String` *`mimeType`*`)` (Returns `dataObject`)

`getStringFromSDO(DataObject` *`sdo`*`, String` *`metaObjectName`*`, String` *`mimeType`*`)` (Returns `String`)

You can access these methods with the Java class `com.ibm.wbi.datahandler.JavaConnectorUtilDH`. This is the class that IBM recommends that you use from a data binding or a Java component. If you have existing code you can use the `AppSide_Connector.JavaConnectorUtil` class.

Usage

The methods defined in the data handler support API can be invoked from either a WebSphere Process Server binding or a Java component. However, because data is typically transformed in the binding in a WebSphere Process Server environment, IBM strongly recommends that the methods of the data handler support API be invoked from a custom data binding rather than a Java component.

Limitations

The data handler support API has the following limitations:

- Binary conversion methods are not supported. That is, no support is provided for getByteArrayFromSDO(), getStreamFromSDO(), getSDO(byte[], and similar calls.

- setEncoding(), setLocale() and setOptions() methods are not exposed via the data handler support API.
- Dynamic child meta objects are not supported.
- You must use WebSphere Business Integration Adapter business object tooling for creation of new objects.

Related reference

Supported WebSphere InterChange Server or WebSphere Business Integration Server Express APIs
In addition to the WebSphere InterChange Server or WebSphere Business Integration Server Express source artifact migration tools provided in WebSphere Process Server and WebSphere Integration Developer, WebSphere Process Server also provides support for many of the APIs that were provided in WebSphere InterChange Server or WebSphere Business Integration Server Express. The migration tools work in conjunction with these WebSphere InterChange Server or WebSphere Business Integration Server Express APIs by preserving your custom snippet code as much as possible when migrating.

Related information

IBM WebSphere Business Integration Adapters/IBM WebSphere InterChange Server Data Handler Guide

Supported WebSphere InterChange Server or WebSphere Business Integration Server Express APIs

In addition to the WebSphere InterChange Server or WebSphere Business Integration Server Express source artifact migration tools provided in WebSphere Process Server and WebSphere Integration Developer, WebSphere Process Server also provides support for many of the APIs that were provided in WebSphere InterChange Server or WebSphere Business Integration Server Express. The migration tools work in conjunction with these WebSphere InterChange Server or WebSphere Business Integration Server Express APIs by preserving your custom snippet code as much as possible when migrating.

Note: These APIs are provided only to support migrated WebSphere InterChange Server or WebSphere Business Integration Server Express applications until they can be modified to use new WebSphere Process Server APIs.
The supported WebSphere InterChange Server or WebSphere Business Integration Server Express APIs are listed below. These APIs provide functions in WebSphere Process Server similar to the function that they provide in WebSphere InterChange Server or WebSphere Business Integration Server Express. See the WebSphere InterChange Server or WebSphere Business Integration Server Express v4.3 documentation for a functional description of these APIs.

CwBiDiEngine
AppSide_Connector/

- BiDiBOTransformation(BusinessObject, String, String, boolean):BusinessObj
- BiDiBusObjTransformation(BusObj, String, String, boolean):BusObj
- BiDiStringTransformation(String, String, String):String

JavaConnectorUtil
AppSide_Connector/

- INFRASTRUCTURE_MESSAGE_FILE
- CONNECTOR_MESSAGE_FILE
- XRD_WARNING

- XRD_TRACE
- XRD_INFO
- XRD_ERROR
- XRD_FATAL
- LEVEL1
- LEVEL2
- LEVEL3
- LEVEL4
- LEVEL5
- createBusinessObject(String):BusinesObjectInterface
- createBusinessObject(String, Locale):BusinesObjectInterface
- createBusinessObject(String, String):BusinesObjectInterface
- createContainer(String):CxObjectContainerInterface
- generateMsg(int, int, int, int, int, Vector):String
- generateMsg(int, int, int, int, Vector):String
- getBlankValue():String
- getEncoding():String
- getIgnoreValue():String
- getLocale():String
- getSDOFromString(String inputString, String sdoName, String metaObjectName, String mimeType)
- getStringFromSDO(DataObject sdo, String metaObjectName, String mimeType)
- isBlankValue(Object):boolean
- isIgnoreValue(Object):boolean
- isTraceEnabled(int):boolean
- logMsg(String)
- logMsg(String, int)
- traceWrite(int, String)

JavaConnectorUtilDH
datahandler/
wbi/
ibm/
com/

- getSDOFromString(String inputString, String sdoName, String metaObjectName, String mimeType)
- getStringFromSDO(DataObject sdo, String metaObjectName, String mimeType)

BusObj
Collaboration/

- BusObj(DataObject)
- BusObj(String)
- BusObj(String, Locale)
- copy(BusObj)
- duplicate():BusObj
- equalKeys(BusObj):boolean
- equals(Object):boolean

- equalsShallow(BusObj):boolean
- exists(String):boolean
- get(int):Object
- get(String):Object
- getBoolean(String):boolean
- getBusObj(String):BusObj
- getBusObjArray(String):BusObjArray
- getCount(String):int
- getDouble(String):double
- getFloat(String):float
- getInt(String):int
- getKeys():String
- getLocale():java.util.Locale
- getLong(String):long
- getLongText(String):String
- getString(String):String
- getType():String
- getValues():String
- getVerb():String
- isBlank(String):boolean
- isKey(String):boolean
- isNull(String):boolean
- isRequired(String):boolean
- keysToString():String
- set(BusObj)
- set(int, Object)
- set(String, boolean)
- set(String, double)
- set(String, float)
- set(String, int)
- set(String, long)
- set(String, Object)
- set(String, String)
- setContent(BusObj)
- setDefaultAttrValues()
- setKeys(BusObj)
- setLocale(java.util.Locale)
- setVerb(String)
- setVerbWithCreate(String, String)
- setWithCreate(String, boolean)
- setWithCreate(String, BusObj)
- setWithCreate(String, BusObjArray)
- setWithCreate(String, double)
- setWithCreate(String, float)
- setWithCreate(String, int)

- setWithCreate(String, long):
- setWithCreate(String, Object)
- setWithCreate(String, String)
- toString():String
- validData(String, boolean):boolean
- validData(String, BusObj):boolean
- validData(String, BusObjArray):boolean
- validData(String, double):boolean
- validData(String, float):boolean
- validData(String, int):boolean
- validData(String, long):boolean
- validData(String, Object):boolean
- validData(String, String):boolean

BusObjArray
Collaboration/

- addElement(BusObj)
- duplicate():BusObjArray
- elementAt(int):BusObj
- equals(BusObjArray):boolean
- getElements():BusObj[]
- getLastIndex():int
- max(String):String
- maxBusObjArray(String):BusObjArray
- maxBusObjs(String):BusObj[]
- min(String):String
- minBusObjArray(String):BusObjArray
- minBusObjs(String):BusObj[]
- removeAllElements()
- removeElement(BusObj)
- removeElementAt(int)
- setElementAt(int, BusObj)
- size():int
- sum(String):double
- swap(int, int)
- toString():String

BaseDLM
DLM/

- BaseDLM(BaseMap)
- getDBConnection(String):CwDBConnection
- getDBConnection(String, boolean):CwDBConnection
- getName():String
- getRelConnection(String):DtpConnection
- implicitDBTransactionBracketing():boolean
- isTraceEnabled(int):boolean

- logError(int)
- logError(int, Object[])
- logError(int, String)
- logError(int, String, String)
- logError(int, String, String, String)
- logError(int, String, String, String, String)
- logError(int, String, String, String, String, String)
- logError(String)
- logInfo(int)
- logInfo(int, Object[])
- logInfo(int, String)
- logInfo(int, String, String)
- logInfo(int, String, String, String)
- logInfo(int, String, String, String, String)
- logInfo(int, String, String, String, String, String)
- logInfo(String)
- logWarning(int)
- logWarning(int, Object[])
- logWarning(int, String)
- logWarning(int, String, String)
- logWarning(int, String, String, String)
- logWarning(int, String, String, String, String)
- logWarning(int, String, String, String, String, String)
- logWarning(String)
- raiseException(RunTimeEntityException)
- raiseException(String, int)
- raiseException(String, int, Object[])
- raiseException(String, int, String)
- raiseException(String, int, String, String)
- raiseException(String, int, String, String, String)
- raiseException(String, int, String, String, String, String)
- raiseException(String, int, String, String, String, String, String)
- raiseException(String, String)
- releaseRelConnection(boolean)
- trace(int, int)
- trace(int, int, Object[])
- trace(int, int, String)
- trace(int, int, String, String)
- trace(int, int, String, String, String)
- trace(int, int, String, String, String, String)
- trace(int, int, String, String, String, String, String)
- trace(int, String)
- trace(String)

CwDBConnection
CwDBConnection/
CxCommon/

- beginTransaction()
- commit()
- executePreparedSQL(String)
- executePreparedSQL(String, Vector)
- executeSQL(String)
- executeSQL(String, Vector)
- executeStoredProcedure(String, Vector)
- getUpdateCount():int
- hasMoreRows():boolean
- inTransaction():boolean
- isActive():boolean
- nextRow():Vector
- release()
- rollback()

CwDBConstants
CwDBConnection/
CxCommon/

- PARAM_IN - 0
- PARAM_INOUT - 1
- PARAM_OUT - 2

CwDBStoredProcedureParam
CwDBConnection/
CxCommon/

- CwDBStoredProcedureParam(int, Array)
- CwDBStoredProcedureParam(int, BigDecimal)
- CwDBStoredProcedureParam(int, boolean)
- CwDBStoredProcedureParam(int, Boolean)
- CwDBStoredProcedureParam(int, byte[])
- CwDBStoredProcedureParam(int, double)
- CwDBStoredProcedureParam(int, Double)
- CwDBStoredProcedureParam(int, float)
- CwDBStoredProcedureParam(int, Float)
- CwDBStoredProcedureParam(int, int)
- CwDBStoredProcedureParam(int, Integer)
- CwDBStoredProcedureParam(int, java.sql.Blob)
- CwDBStoredProcedureParam(int, java.sql.Clob)
- CwDBStoredProcedureParam(int, java.sql.Date)
- CwDBStoredProcedureParam(int, java.sql.Struct)
- CwDBStoredProcedureParam(int, java.sql.Time)
- CwDBStoredProcedureParam(int, java.sql.Timestamp)
- CwDBStoredProcedureParam(int, Long)
- CwDBStoredProcedureParam(int, String)

- CwDBStoredProcedureParam(int, String, Object)
- getParamType():int getValue():Object

DataHandler (Abstract Class)
DataHandlers/
crossworlds/
com/

- createHandler(String, String, String):DataHandler
- getBO(InputStream, Object):BusinessObjectInterface
- getBO(Object, BusinessObjectInterface, Object)
- getBO(Object, Object):BusinessObjectInterface
- getBO(Reader, BusinessObjectInterface, Object) (Abstract Method)
- getBO(Reader, Object):BusinessObjectInterface (Abstract Method)
- getBO(String, Object):BusinessObjectInterface
- getBOName(InputStream):String
- getBOName(Reader):String
- getBOName(String):String
- getBooleanOption(String):boolean
- getEncoding():String
- getLocale():Locale
- getOption(String):String
- getStreamFromBO(BusinessObjectInterface, Object):InputStream (Abstract Method)
- getStringFromBO(BusinessObjectInterface, Object):String (Abstract Method)
- setConfigMOName(String)
- setEncoding(String)
- setLocale(Locale)
- setOption(String, String)
- traceWrite(String, int)

NameHandler (Abstract Class)
DataHandlers/
crossworlds/
com/

- getBOName(Reader, String):String) (Abstract Method)

ConfigurationException (extends java.lang.Exception)
Exceptions/
DataHandlers/
crossworlds/
com/

MalformedDataException (extends java.lang.Exception)
Exceptions/
DataHandlers/
crossworlds/
com/

NotImplementedException (extends java.lang.Exception)
Exceptions/

**DataHandlers/
crossworlds/
com/**

**BusinessObjectInterface
CxCommon/**

- clone():Object
- dump():String
- getAppText():String
- getAttrCount():int
- getAttrDesc(int):CxObjectAttr
- getAttrDesc(String):CxObjectAttr
- getAttribute(String):Object
- getAttributeIndex(String):int
- getAttributeType(int):int
- getAttributeType(String):int
- getAttrName(int):String
- getAttrValue(int):Object
- getAttrValue(String):Object
- getBusinessObjectVersion():String
- getDefaultAttrValue(int):String
- getDefaultAttrValue(String):String
- getLocale():String
- getName():String
- getParentBusinessObject():BusinessObjectInterface
- getVerb():String
- getVerbAppText(String):String
- isBlank(int):boolean
- isBlank(String):boolean
- isIgnore(int):boolean
- isIgnore(String):boolean
- isVerbSupported(String):boolean
- makeNewAttrObject(int):Object
- makeNewAttrObject(String):Object
- setAttributeWithCreate(String, Object)
- setAttrValue(int, Object)
- setAttrValue(String, Object)
- setDefaultAttrValues()
- setLocale(Locale)
- setLocale(String)
- setVerb(String)

**CxObjectAttr
CxCommon/**

- BOOLEAN
- BOOLSTRING

- DATE
- DATESTRING
- DOUBLE
- DOUBSTRING
- FLOAT
- FLTSTRING
- INTEGER
- INTSTRING
- INVALID_TYPE_NUM
- INVALID_TYPE_STRING
- LONGTEXT
- LONGTEXTSTRING
- MULTIPLECARDSTRING
- OBJECT
- SINGLECARDSTRING
- STRING
- STRSTRING
- equals(Object):boolean
- getAppText():String
- getCardinality():String
- getDefault():String
- getMaxLength():int
- getName():String
- getRelationType():String
- getTypeName():String
- getTypeNum():String
- hasCardinality(String):boolean
- hasName(String):boolean
- hasType(String):boolean
- isForeignKeyAttr():boolean
- isKeyAttr():boolean
- isMultipleCard():boolean
- isObjectType():boolean
- isRequiredAttr():boolean
- isType(Object):boolean

CxObjectContainerInterface
CxCommon/

- getBusinessObject(int):BusinessObjectInterface
- getObjectCount():int
- insertBusinessObject(BusinessObjectInterface)
- removeAllObjects()
- removeBusinessObjectAt(int)
- setBusinessObject(int, BusinessObjectInterface)

DtpConnection
Dtp/
CxCommon/

- beginTran()
- commit()
- executeSQL(String)
- executeSQL(String, Vector)
- executeStoredProcedure(String, Vector)
- getUpdateCount():int
- hasMoreRows():boolean
- inTransaction():boolean
- isActive():boolean
- nextRow():Vector
- rollback()

DtpDataConversion
Dtp/
CxCommon/

- BOOL_TYPE - 4
- CANNOTCONVERT - 2
- DATE_TYPE - 5
- DOUBLE_TYPE - 3
- FLOAT_TYPE - 2
- INTEGER_TYPE - 0
- LONGTEXT_TYPE - 6
- OKTOCONVERT - 0
- POTENTIALDATALOSS - 1
- STRING_TYPE - 1
- UNKNOWN_TYPE - 999
- getType(double):int
- getType(float):int
- getType(int):int
- getType(Object):int
- isOKToConvert(int, int):int
- isOKToConvert(String, String):int
- toBoolean(boolean):Boolean
- toBoolean(Object):Boolean
- toDouble(double):Double
- toDouble(float):Double
- toDouble(int):Double
- toDouble(Object):Double
- toFloat(double):Float
- toFloat(float):Float
- toFloat(int):Float
- toFloat(Object):Float
- toInteger(double):Integer

- toInteger(float):Integer
- toInteger(int):Integer
- toInteger(Object):Integer
- toPrimitiveBoolean(Object):boolean
- toPrimitiveDouble(float):double
- toPrimitiveDouble(int):double
- toPrimitiveDouble(Object):double
- toPrimitiveFloat(double):float
- toPrimitiveFloat(int):float
- toPrimitiveFloat(Object):float
- toPrimitiveInt(double):int
- toPrimitiveInt(float):int
- toPrimitiveInt(Object):int
- toString(double):String
- toString(float):String
- toString(int):String
- toString(Object):String

DtpDate
Dtp/
CxCommon/

- DtpDate()
- DtpDate(long, boolean)
- DtpDate(String, String)
- DtpDate(String, String, String[], String[])
- addDays(int):DtpDate
- addMonths(int):DtpDate
- addWeekdays(int):DtpDate
- addYears(int):DtpDate
- after(DtpDate):boolean
- before(DtpDate):boolean
- calcDays(DtpDate):int
- calcWeekdays(DtpDate):int
- get12MonthNames():String[]
- get12ShortMonthNames():String[]
- get7DayNames():String[]
- getCWDate():String
- getDayOfMonth():String
- getDayOfWeek():String
- getHours():String
- getIntDay():int
- getIntDayOfWeek():int
- getIntHours():int
- getIntMilliSeconds():int
- getIntMinutes():int
- getIntMonth():int

- getIntSeconds():int
- getIntYear():int
- getMaxDate(BusObjArray, String, String):DtpDate
- getMaxDateBO(BusObj[], String, String):BusObj[]
- getMaxDateBO(BusObjArray, String, String):BusObj[]
- getMinDate(BusObjArray, String, String):DtpDate
- getMinDateBO(BusObj[], String, String):BusObj[]
- getMinDateBO(BusObjArray, String, String):BusObj[]
- getMinutes():String
- getMonth():String
- getMSSince1970():long
- getNumericMonth():String
- getSeconds():String
- getShortMonth():String
- getYear():String
- set12MonthNames(String[], boolean)
- set12MonthNamesToDefault()
- set12ShortMonthNames(String[])
- set12ShortMonthNamesToDefault()
- set7DayNames(String[])
- set7DayNamesToDefault()
- toString():String
- toString(String):String
- toString(String, boolean):String

DtpMapService
Dtp/
CxCommon/

- runMap(String, String, BusObj[], CxExecutionContext):BusObj[]

DtpSplitString
Dtp/
CxCommon/

- DtpSplitString(String, String)
- elementAt(int):String
- firstElement():String
- getElementCount():int
- getEnumeration():Enumeration
- lastElement():String
- nextElement():String
- prevElement():String
- reset()

DtpUtils
Dtp/
CxCommon/

- padLeft(String, char, int):String

- padRight(String, char, int):String
- stringReplace(String, String, String):String
- truncate(double):int
- truncate(double, int):double
- truncate(float):int
- truncate(float, int):double
- truncate(Object):int
- truncate(Object, int):double

BusObjInvalidVerbException (extends InterchangeExceptions)
Exceptions/
CxCommon/

- getFormattedMessage()

IdentityRelationship
relationship/
utilities/
crossworlds/
com/

- addMyChildren(String, String, BusObj, String, Object, CxExecutionContext)
- deleteMyChildren(String, String, BusObj, String, CxExecutionContext)
- deleteMyChildren(String, String, BusObj, String, Object, CxExecutionContext)
- foreignKeyLookup(String, String, BusObj, String, BusObj, String, CxExecutionContext)
- foreignKeyXref(String, String, String, BusObj, String, BusObj, String, CxExecutionContext)
- maintainChildVerb(String, String, String, BusObj, String, BusObj, String, CxExecutionContext, boolean, boolean)
- maintainCompositeRelationship(String, String, BusObj, Object, CxExecutionContext)
- maintainSimpleIdentityRelationship(String, String, BusObj, BusObj, CxExecutionContext)
- updateMyChildren(String, String, BusObj, String, String, String, String, CxExecutionContext)

MapExeContext
Dtp/
CxCommon/

- ACCESS_REQUEST - ″SUBSCRIPTION_DELIVERY″
- ACCESS_RESPONSE - ″ACCESS_RETURN_REQUEST″
- EVENT_DELIVERY - ″SUBSCRIPTION_DELIVERY″
- SERVICE_CALL_FAILURE - ″CONSUME_FAILED″
- SERVICE_CALL_REQUEST - ″CONSUME″
- SERVICE_CALL_RESPONSE - ″DELIVERBUSOBJ″
- getConnName():String
- getGenericBO():BusObj
- getInitiator():String
- getLocale():java.util.Locale
- getOriginalRequestBO():BusObj

- setConnName(String)
- setInitiator(String)
- setLocale(java.util.Locale)

Participant RelationshipServices/ Server/

- Participant(String, String, int, BusObj)
- Participant(String, String, int, String)
- Participant(String, String, int, long)
- Participant(String, String, int, int)
- Participant(String, String, int, double)
- Participant(String, String, int, float)
- Participant(String, String, int, boolean)
- Participant(String, String, BusObj)
- Participant(String, String, String)
- Participant(String, String, long)
- Participant(String, String, int)
- Participant(String, String, double)
- Participant(String, String, float)
- Participant(String, String, boolean)
- getBoolean():boolean
- getBusObj():BusObj
- getDouble():double
- getFloat():float
- getInstanceId():int
- getInt():int
- getLong():long
- getParticipantDefinition():String
- getRelationshipDefinition():String
- getString():String INVALID_INSTANCE_ID
- set(boolean)
- set(BusObj)
- set(double)
- set(float)
- set(int)
- set(long)
- set(String)
- setInstanceId(int)
- setParticipantDefinition(String)
- setRelationshipDefinition(String)
- setParticipantDefinition(String)
- setRelationshipDefinition(String)

Relationship
RelationshipServices/
Server/

- addMyChildren(String, String, BusObj, String, Object, CxExecutionContext)
- addParticipant(Participant):int
- addParticipant(String, String, boolean):int
- addParticipant(String, String, BusObj):int
- addParticipant(String, String, double):int
- addParticipant(String, String, float):int
- addParticipant(String, String, int):int
- addParticipant(String, String, int, boolean):int
- addParticipant(String, String, int, BusObj):int
- addParticipant(String, String, int, double):int
- addParticipant(String, String, int, float):int
- addParticipant(String, String, int, int):int
- addParticipant(String, String, int, long):int
- addParticipant(String, String, int, String):int
- addParticipant(String, String, long):int
- addParticipant(String, String, String):int
- create(Participant):int
- create(String, String, boolean):int
- create(String, String, BusObj):int
- create(String, String, double):int
- create(String, String, float):int
- create(String, String, int):int
- create(String, String, long):int
- create(String, String, String):int
- deactivateParticipant(Participant)
- deactivateParticipant(String, String, boolean)
- deactivateParticipant(String, String, BusObj)
- deactivateParticipant(String, String, double)
- deactivateParticipant(String, String, float)
- deactivateParticipant(String, String, int)
- deactivateParticipant(String, String, long)
- deactivateParticipant(String, String, String)
- deactivateParticipantByInstance(String, String, int)
- deactivateParticipantByInstance(String, String, int, boolean)
- deactivateParticipantByInstance(String, String, int, BusObj)
- deactivateParticipantByInstance(String, String, int, double)
- deactivateParticipantByInstance(String, String, int, float)
- deactivateParticipantByInstance(String, String, int, int)
- deactivateParticipantByInstance(String, String, int, long)
- deactivateParticipantByInstance(String, String, int, String)
- deleteMyChildren(String, String, BusObj, String, CxExecutionContext)
- deleteMyChildren(String, String, BusObj, String, Object, CxExecutionContext)

- deleteParticipant(Participant)
- deleteParticipant(String, String, boolean)
- deleteParticipant(String, String, BusObj)
- deleteParticipant(String, String, double)
- deleteParticipant(String, String, float)
- deleteParticipant(String, String, int)
- deleteParticipant(String, String, long)
- deleteParticipant(String, String, String)
- deleteParticipantByInstance(String, String, int)
- deleteParticipantByInstance(String, String, int, boolean)
- deleteParticipantByInstance(String, String, int, BusObj)
- deleteParticipantByInstance(String, String, int, double)
- deleteParticipantByInstance(String, String, int, float)
- deleteParticipantByInstance(String, String, int, int)
- deleteParticipantByInstance(String, String, int, long)
- deleteParticipantByInstance(String, String, int, String)
- getNewID(String):int
- maintainCompositeRelationship(String, String, BusObj, Object, CxExecutionContext)
- maintainSimpleIdentityRelationship(String, String, BusObj, BusObj, CxExecutionContext)
- retrieveInstances(String, boolean):int[]
- retrieveInstances(String, BusObj):int[]
- retrieveInstances(String, double):int[]
- retrieveInstances(String, float):int[]
- retrieveInstances(String, int):int[]
- retrieveInstances(String, long):int[]
- retrieveInstances(String, String):int[]
- retrieveInstances(String, String, boolean):int[]
- retrieveInstances(String, String, BusObj):int[]
- retrieveInstances(String, String, double):int[]
- retrieveInstances(String, String, float):int[]
- retrieveInstances(String, String, int):int[]
- retrieveInstances(String, String, long):int[]
- retrieveInstances(String, String, String):int[]
- retrieveInstances(String, String[], boolean):int[]
- retrieveInstances(String, String[], BusObj):int[]
- retrieveInstances(String, String[], double):int[]
- retrieveInstances(String, String[], float):int[]
- retrieveInstances(String, String[], int):int[]
- retrieveInstances(String, String[], long):int[]
- retrieveInstances(String, String[], String):int[]
- retrieveParticipants(String):Participant[]
- retrieveParticipants(String, String):Participant[]
- retrieveParticipants(String, String[]):Participant[]

- retrieveParticipants(String, int):Participant[]
- retrieveParticipants(String, String, int):Participant[]
- retrieveParticipants(String, String[], int):Participant[]
- updateMyChildren(String, String, BusObj, String, String, String, String, CxExecutionContext)
- updateParticipant(String, String, BusObj)
- updateParticipantByInstance(Participant)
- updateParticipantByInstance(String, String, int)
- updateParticipantByInstance(String, String, int, BusObj)

UserStoredProcedureParam
Dtp/
CxCommon/

- UserStoredProcedureParam(int, String, Object, String, String)
- getParamDataTypeJavaObj():String
- getParamDataTypeJDBC():int
- getParamIndex():int
- getParamIOType():String
- getParamName():String
- getParamValue():Object
- setParamDataTypeJavaObj(String)
- setParamDataTypeJDBC(int)
- setParamIndex(int)
- setParamIOType(String)
- setParamName(String)
- setParamValue(Object)
- PARAM_TYPE_IN - ″IN″
- PARAM_TYPE_OUT - ″OUT″
- PARAM_TYPE_INOUT - ″INOUT″
- DATA_TYPE_STRING - ″String″
- DATA_TYPE_INTEGER - ″Integer″
- DATA_TYPE_DOUBLE - ″Double″
- DATA_TYPE_FLOAT - ″Float″
- DATA_TYPE_BOOLEAN - ″Boolean″
- DATA_TYPE_TIME - ″java.sql.Time″
- DATA_TYPE_DATE - ″java.sql.Date″
- DATA_TYPE_TIMESTAMP - ″java.sql.Timestamp″
- DATA_TYPE_BIG_DECIMAL - ″java.math.BigDecimal″
- DATA_TYPE_LONG_INTEGER - ″Long″
- DATA_TYPE_BINARY - ″byte[]″
- DATA_TYPE_CLOB - ″Clob″
- DATA_TYPE_BLOB - ″Blob″
- DATA_TYPE_ARRAY - ″Array″
- DATA_TYPE_STRUCT - ″Struct″
- DATA_TYPE_REF - ″Ref″

BaseCollaboration
Collaboration/

- BaseCollaboration(com.ibm.bpe.api.ProcessInstanceData)
- AnyException - ″AnyException″
- AppBusObjDoesNotExist - ″BusObjDoesNotExist″
- AppLogOnFailure - ″AppLogOnFailure″
- AppMultipleHits - ″AppMultipleHits″
- AppRequestNotYetSent - ″AppRequestNotYetSent″
- AppRetrieveByContentFailed - ″AppRetrieveByContent″
- AppTimeOut - ″AppTimeOut″
- AppUnknown - ″AppUnknown″
- AttributeException - ″AttributeException″
- existsConfigProperty(String):boolean
- getConfigProperty(String):String
- getConfigPropertyArray(String):String[]
- getCurrentLoopIndex():int
- getDBConnection(String):CwDBConnection
- getDBConnection(String, boolean):CwDBConnection getLocale():java.util.Locale
- getMessage(int):String
- getMessage(int, Object[]):String
- getName():String
- implicitDBTransactionBracketing():boolean
- isCallerInRole(String):boolean
- isTraceEnabled(int):boolean
- JavaException - ″JavaException″
- logError(int)
- logError(int, Object[])
- logError(int, String)
- logError(int, String, String)
- logError(int, String, String, String)
- logError(int, String, String, String, String)
- logError(int, String, String, String, String, String)
- logError(String)
- logInfo(int)
- logInfo(int, Object[])
- logInfo(int, String)
- logInfo(int, String, String)
- logInfo(int, String, String, String)
- logInfo(int, String, String, String, String)
- logInfo(int, String, String, String, String, String)
- logInfo(String)
- logWarning(int)
- logWarning(int, Object[])
- logWarning(int, String)
- logWarning(int, String, String)

- logWarning(int, String, String, String)
- logWarning(int, String, String, String, String)
- logWarning(int, String, String, String, String, String)
- logWarning(String)
- not(boolean):boolean ObjectException - ″ObjectException″
- OperationException - ″OperationException″
- raiseException(CollaborationException)
- raiseException(String, int)
- raiseException(String, int, Object[])
- raiseException(String, int, String)
- raiseException(String, int, String, String)
- raiseException(String, int, String, String, String)
- raiseException(String, int, String, String, String, String)
- raiseException(String, int, String, String, String, String, String)
- raiseException(String, String)
- ServiceCallException - ″ConsumerException″
- ServiceCallTransportException - ″ServiceCallTransportException″
- SystemException - ″SystemException″
- trace(int, int)
- trace(int, int, Object[])
- trace(int, int, String)
- trace(int, int, String, String)
- trace(int, int, String, String, String)
- trace(int, int, String, String, String, String)
- trace(int, int, String, String, String, String, String)
- trace(int, String)
- trace(String)
- TransactionException - ″TransactionException″

CxExecutionContext
CxCommon/

- CxExecutionContext()
- getContext(String):Object
- MAPCONTEXT - ″MAPCONTEXT″
- setContext(String, Object)

CollaborationException
Collaboration/

- getMessage():String
- getMsgNumber():int
- getSubType():String
- getText():String
- getType():String
- toString():String

Filter
crossworlds/
com/

- Filter(BaseCollaboration)
- filterExcludes(String, String):boolean
- filterIncludes(String, String):boolean
- recurseFilter(BusObj, String, boolean, String, String):boolean
- recursePreReqs(String, Vector):int

Globals
crossworlds/
com/

- Globals(BaseCollaboration)
- callMap(String, BusObj):BusObj

SmartCollabService
crossworlds/
com/

- SmartCollabService()
- SmartCollabService(BaseCollaboration)
- doAgg(BusObj, String, String, String):BusObj
- doMergeHash(Vector, String, String):Vector
- doRecursiveAgg(BusObj, String, String, String):BusObj
- doRecursiveSplit(BusObj, String):Vector
- doRecursiveSplit(BusObj, String, boolean):Vector
- getKeyValues(BusObj, String):String
- merge(Vector, String):BusObj
- merge(Vector, String, BusObj):BusObj
- split(BusObj, String):Vector

StateManagement
crossworlds/
com/

- StateManagement()
- beginTransaction()
- commit()
- deleteBO(String, String, String)
- deleteState(String, String, String, int)
- persistBO(String, String, String, String, BusObj)
- recoverBO(String, String, String):BusObj
- releaseDBConnection()
- resetData()
- retrieveState(String, String, String, int):int
- saveState(String, String, String, String, int, int, double)
- setDBConnection(CwDBConnection)
- updateBO(String, String, String, String, BusObj)
- updateState(String, String, String, String, int, int)

EventKeyAttrDef EventManagement/ CxCommon/

- EventKeyAttrDef()
- EventKeyAttrDef(String, String)
- public String keyName
- public String keyValue

EventQueryDef EventManagement/ CxCommon/

- EventQueryDef()
- EventQueryDef(String, String, String, String, int)
- public String nameConnector
- public String nameCollaboration
- public String nameBusObj
- public String verb
- public int ownerType

FailedEventInfo EventManagement/ CxCommon/

- FailedEventInfo()
- FailedEventInfo(String x6, int, EventKeyAttrDef[], int, int, String, String, int)
- public String nameOwner
- public String nameConnector
- public String nameBusObj
- public String nameVerb
- public String strTime
- public String strMessage
- public int wipIndex
- public EventKeyAttrDef[] strbusObjKeys
- public int nKeys
- public int eventStatus
- public String expirationTime
- public String scenarioName
- public int scenarioState

Limitations when migrating from WebSphere InterChange Server or WebSphere Business Integration Server Express

Some characteristics of WebSphere InterChange Server or WebSphere Business Integration Server Express are not precisely duplicated by WebSphere Process Server. Therefore you might have to modify your applications after migration to get them to perform as they did in WebSphere InterChange Server or WebSphere Business Integration Server Express.

The following sections describe these limitations and possible solutions.

Transaction levels

There is no direct mapping of the levels of transaction between WebSphere InterChange Server or WebSphere Business Integration Server Express collaborations and WebSphere Process Server BPEL files. Therefore, the transaction level specified in the WebSphere InterChange Server or WebSphere Business Integration Server Express collaboration is ignored and the default BPEL transaction level will be used in the migrated application. You should understand BPEL transactions and adjust your migrated applications accordingly to get the desired functionality.

Note: Pending transactions will not be migrated. All transactions should be concluded before starting migration.

Compensation

WebSphere Process Server compensation is different from WebSphere InterChange Server or WebSphere Business Integration Server Express compensation. You should evaluate the new types of compensation offered by WebSphere Process Server and choose the type that best suits your application.

Event Summary and Change Summary not supported when using WebSphere InterChange Server or WebSphere Business Integration Server Express APIs on WebSphere Process Server

Problem: Event Summary and Change Summary do not contain expected information in migrated WebSphere InterChange Server or WebSphere Business Integration Server Express applications. **Cause:** Business Objects (BusObjs) in WebSphere InterChange Server or WebSphere Business Integration Server Express do not support Change Summary and Event Summary. The WebSphere InterChange Server or WebSphere Business Integration Server Express APIs supported in WebSphere Process Server work with the WebSphere InterChange Server or WebSphere Business Integration Server Express type BusObj, so any use of those APIs forces a conversion into a BusObj. When this happens, any Event Summary and Change Summary information contained in a WebSphere Process Server DataObject that is converted into a BusObj is lost. Applications generated by migration from WebSphere InterChange Server or WebSphere Business Integration Server Express will use the WebSphere InterChange Server or WebSphere Business Integration Server Express APIs in WebSphere Process Server, so Event Summary and Change Summary cannot be used with these applications until the code is manually updated to stop using any of the WebSphere InterChange Server or WebSphere Business Integration Server Express APIs. **Solution:** Remove all use of WebSphere InterChange Server or WebSphere Business Integration Server Express APIs or change them to WebSphere Process Server APIs.

Related concepts

Troubleshooting migration from WebSphere InterChange Server or WebSphere Business Integration Server Express
Find solutions to problems you encounter with migration as well as instructions for turning on logging and tracing.

Related reference

Postmigration considerations
When applications have been migrated from WebSphere InterChange Server or WebSphere Business Integration Server Express to WebSphere Process Server, special attention is required in some areas to enable migrated applications to function in WebSphere Process Server consistently with their intended function due to differences between the architectures of WebSphere Process Server and WebSphere InterChange Server or WebSphere Business Integration Server Express.

Related information

Compensation in business processes

Troubleshooting migration from WebSphere InterChange Server or WebSphere Business Integration Server Express

Find solutions to problems you encounter with migration as well as instructions for turning on logging and tracing.

Related concepts

Limitations when migrating from WebSphere InterChange Server or WebSphere Business Integration Server Express
Some characteristics of WebSphere InterChange Server or WebSphere Business Integration Server Express are not precisely duplicated by WebSphere Process Server. Therefore you might have to modify your applications after migration to get them to perform as they did in WebSphere InterChange Server or WebSphere Business Integration Server Express.

Related reference

Postmigration considerations
When applications have been migrated from WebSphere InterChange Server or WebSphere Business Integration Server Express to WebSphere Process Server, special attention is required in some areas to enable migrated applications to function in WebSphere Process Server consistently with their intended function due to differences between the architectures of WebSphere Process Server and WebSphere InterChange Server or WebSphere Business Integration Server Express.

Premigration considerations
Consider these guidelines for the development of integration artifacts for WebSphere InterChange Server or WebSphere Business Integration Server Express to ease the migration of WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts to WebSphere Process Server.

Enabling logging and tracing for supported WebSphere InterChange Server or WebSphere Business Integration Server Express APIs

Enable logging and tracing for the supported WebSphere InterChange Server or WebSphere Business Integration Server Express APIs through the administrative console.

About this task

If your migrated application includes any supported WebSphere InterChange Server or WebSphere Business Integration Server Express APIs, you can enable logging and tracing for them for troubleshooting purposes.

Procedure

1. Launch the administrative console.
2. From the left (navigation) panel, select **Troubleshooting > Logs and Trace**.
3. In the right panel, select the name of the server on which you want to enable logging and tracing.
4. In the right panel, under ″General properties,″ select **Change Log Level Details**.
5. Select the Runtime tab. (Selecting the Runtime tab allows you to make this change in real time without requiring you to restart the server.)
6. Add the name of the package followed by `=all` to the list of logged packages in the box on the screen. Separate this new entry from any existing entries with a colon. For example, `CxCommon=all`. In this case, `CxCommon` is the name of the package for a set of supported WebSphere InterChange Server or WebSphere Business Integration Server Express APIs. Specifying `all` enables all logging and tracing. See Supported WebSphere InterChange Server APIs for a list of the APIs, including their package names.
7. Select **Apply**.
8. To keep this configuration after the server is restarted, select the **Save runtime changes to configuration as well** check box.
9. Select **OK**.
10. When the next screen appears, select **Save** to save your changes.

Related information

Supported WebSphere InterChange Server APIs

Failure trying to serialize an object that is not serializable in a migrated BPEL file

If a serialization failure occurs in a BPEL file generated by the migration, you might be able to modify it to prevent the failure from occurring.

Problem: A serialization failure occurs in a custom snippet node of a business process execution language (BPEL) file generated by migration because an attempt is made to serialize an object that is not serializable.

Cause: In WebSphere InterChange Server or WebSphere Business Integration Server Express, a Collaboration Template is compiled into a single Java class. In WebSphere Process Server, each node in a BPEL file might compile into a separate Java class. In WebSphere InterChange Server or WebSphere Business Integration Server Express, a variable can be declared once and shared throughout the various steps of a Collaboration Template. To simulate that behavior in the migrated BPEL file, each variable used in a code snippet must be retrieved at the start of the snippet and saved at the end of the snippet. Variables defined in WebSphere InterChange Server or WebSphere Business Integration Server Express Port definitions become BPEL variables. These are retrieved into BusObj variables at the beginning of each snippet (if referenced in the snippet) and saved back to the BPEL variables at the end of each snippet. For example, a retrieval at the beginning of snippets looks like this:

```
BusObj tempBusObj = null;if (tempBusObj_var != null) { tempBusObj =
      new BusObj(tempBusObj_var); };
```

and a save at the end of snippets looks like this:

```
if (tempBusObj == null) { tempBusObj_var = null; } else { tempBusObj_var =
      tempBusObj.getBusinessGraph(); }
```

Other variables used in the WebSphere InterChange Server or WebSphere Business Integration Server Express snippet code are serialized and stored as a String in a BPEL variable named *CollabTemplateName_var*. These variables are deserialized at the beginning of each BPEL snippet, and then serialized and saved at the end of each BPEL Snippet that they are referenced in. For example, objects are retrieved like this:

```
BusObj tempBusObj = (BusObj)BaseCollaboration.deserialize
  (FrontEndCollab_var.getString("tempBusObj"));
```

and objects are saved like this:

```
FrontEndCollab_var.setString("tempBusObj", BaseCollaboration.serialize(tempBusObj));
```

If the type of the object being serialized is not serializable, then using serialize and deserialize will fail when the BPEL is run.

Solution: After migration, modify the BPEL file as follows:

- For any variable that is not Java-serializable, update the BPEL snippets to remove the serialization and deserialization statements. If the variable needs to be shared across snippets (instead of being re-created in each snippet) another method must be used to preserve the value of the variable across snippets.
- Manually define BPEL variables for variables of type BusObj that are not declared in the WebSphere InterChange Server or WebSphere Business Integration Server Express Port definitions but are used on Partner Invokes. This is a manual step because variables used during invokes in WebSphere Process Server must be strongly typed, and the migration tools cannot accurately determine that type from the WebSphere InterChange Server or WebSphere Business Integration Server Express snippets.

  Note: The naming convention used by the migration tools is to add `_var` to the name of the variable in the snippet code when naming the BPEL variables. For example, for a variable called tempBusObj in the snippet code, the migration tools will create a BPEL variable named `tempBusObj_var`.
- For variables that must be declared manually as BPEL variables, change the BPEL snippet code so that it uses the ″deserialize/serialize″ method of preserving these variables rather than the ″retrieve from/store into BPEL variable″ method of preserving these variables.

New behavior for heritage APIs in WebSphere Process Server, version 6.2

In version 6.2 of WebSphere Process Server, heritage APIs use WebSphere Process Server Service Data Objects to store attribute states and data that were formerly stored by the BusinessObjectInterface interface. As a result, the behavior of some method calls in the BusinessObjectInterface and CxObjectContainerInterface interfaces have changed.

The major change to the heritage APIs (HAPIs) in WebSphere Process Server version 6.2 is that the WebSphere InterChange Server BusinessObjectInterface

interface is no longer the root storage object for HAPI. Instead, a WebSphere Process Server Service Data Object (SDO) is now used to store attribute states and data.

If you use the Java equivalence operator and weakly typed attribute principles, the behavior of the method calls in the BusinessObjectInterface and CxObjectContainerInterface interfaces is different, as described in the following sections:

- "Using the Java equivalence operator when performing a Set operation followed by a Get operation"
- "Using the Java equivalence operator when setting a BusinessObjectInterface object to more than one target attribute" on page 198
- "Using the Java equivalence operator when you set and retrieve a BusinessObjectInterface object from the CxObjectContainerInterface interface" on page 199
- "Using weakly typed attribute data types for the BusObj class validData methods" on page 200

Using the Java equivalence operator when performing a Set operation followed by a Get operation

A different BusinessObjectInterface object is returned when performing a Set operation followed by a Get operation of a BusinessObjectInterface object for one target attribute. The following table describes the previous behavior, current behavior, and an example of what to change if you previously had used the Java equivalence operator when performing a Set operations followed by a Get operation.

Table 6. Behavior changes: Using the Java equivalence operator with Set and Get operations

Behavior type	Description
Behavior prior to WebSphere Process Server version 6.2	The same BusinessObjectInterface container that was set was also retrieved, and you could use the Java equivalence operator "==" to determine if they were the same. Example: `boolean b = (JavaObjectA == JavaObjectB)`
Behavior after WebSphere Process Server version 6.2	The original BusinessObjectInterface container is discarded, and when you perform a Get operation to retrieve the BusinessObjectInterface object, a new container is created. The returned container is not the same object, but the root object that it wraps is the same object. A new method, isEquivalent, has been added to the BusinessObjectInterface class: BusinessObjectInterface.isEquivalent(BOI). When you want to determine if the two BusinessObjectInterface objects are equivalent, use the isEquivalent method to perform the comparison.

Table 6. Behavior changes: Using the Java equivalence operator with Set and Get operations (continued)

Behavior type	Description

Example of new behavior	The following example shows the use of isEquivalent. You have a BusinessObjectInterface object of type MasterBusinessObject with the attribute Attr_Nine, which is a BusinessObjectInterface object of type HelloWorld:
	`BusinessObjectInterface mboBOI, hw1BOI, hw2BOI;`
	`hw1BOI.setAttrValue("Message", "hw1BOI_message");`
	`hw1BOI.setVerb("Create");`
	`mboBOI.setAttrValue("Attr_Nine", hw1BOI);`
	`hw2BOI = mboBOI.getAttrValue("Attr_Nine");`
	Instead of:
	`boolean result = (hw1BOI == hw2BOI); assertTrue(result);`
	Use this:
	`boolean result = hw1BOI.isEquivalent(hw2BOI); assertTrue(result);`

Using the Java equivalence operator when setting a BusinessObjectInterface object to more than one target attribute

Setting a BusinessObjectInterface object to more than one target attribute sets a cloned object. This applies both to elements of a BusObjArray class and to multiple target attributes. The following table describes the previous behavior, current behavior, and an example of what to change if you previously had used the Java equivalence operator when setting a BusinessObjectInterface object to more than one target attribute.

Table 7. Behavior changes: Using the Java equivalence operator with more than one target attribute

Behavior type	Description
Behavior prior to WebSphere Process Server version 6.2	You could set a BusinessObjectInterface object to multiple locations, and all of the locations contained a reference to the original BusinessObjectInterface object. If you changed the attribute in one BusinessObjectInterface object, that change was reflected in all of the other references for that object.
Behavior after WebSphere Process Server version 6.2	Service Data Object (SDO) rules prevent you from setting the same SDO to more than one target property. If you try to set the SDO to more than one target property, the SDO moves from one attribute to the next, leaving a ″null″ value at the previous attribute location. Now, instead of leaving a ″null″ value when the BusinessObjectInterface object is set to a second, third, and so on, location, the object is cloned into the multiple locations. For example, you have a BusinessObjectInterface object of the type MasterBusinessObject, with the attributes Attr_Nine and Attr_Eleven that are of the type HelloWorld. If you set the same HelloWorld object to both attributes, then Attr_Nine is assigned the original object, and Attr_Eleven is assigned a clone. The clone is a snapshot of the object at the time that it is cloned. If you want to determine if two BusinessObjectInterface objects are equivalent, do not use the Java equivalence operator; instead, use the isEquivalent method to perform the comparison.

Table 7. Behavior changes: Using the Java equivalence operator with more than one target attribute (continued)

Behavior type	Description

Example of new behavior	The following example shows the use of isEquivalent and clones. You have a BusinessObjectInterface object of type MasterBusinessObject, with the attributes Attr_Nine and Attr_Eleven that are of type HelloWorld:
	`BusinessObjectInterface mboBOI;`
	`BusinessObjectInterface hw1BOI, hw2BOI, hw3BOI;`
	`hw1BOI.setAttrValue("Message", "hw1BOI_message");`
	`hw1BOI.setVerb("Create");`
	`mboBOI.setAttrValue("Attr_Nine", hw1BOI);`
	`mboBOI.setAttrValue("Attr_Eleven", hw1BOI);`
	`hw2BOI = mboBOI.getAttrValue("Attr_Nine");`
	`hw3BOI = mboBOI.getAttrValue("Attr_ Eleven ");`
	Instead of:
	`boolean result = hw2BOI == hw3BOI); assertTrue(result);`
	Use isEquivalent instead:
	`boolean result = hw2BOI.isEquivalent(hw3BOI); assertTrue(result);`
	The cloned objects do not share a reference, and changes to the original BusinessObjectInterface object are not reflected in the cloned BusinessObjectInterface object:
	`hw1BOI.setAttrValue("Message", "hw1BOI_message changed");`
	`boolean result = hw1BOI.isEquivalent(hw2BOI);`
	`assertTrue(result);`
	`boolean result = hw1BOI.isEquivalent(hw3BOI);`
	`assertFalse(result);`
	`boolean result = hw2BOI.isEquivalent(hw3BOI);`
	`assertFalse(result);`

Using the Java equivalence operator when you set and retrieve a BusinessObjectInterface object from the CxObjectContainerInterface interface

The following table describes the previous behavior, current behavior, and an example of what to change if you previously had used the Java equivalence operator when setting and retrieving a BusinessObjectInterface object from the CxObjectContainerInterface interface.

Table 8. Behavior changes: Using the Java equivalence operator with the CxObjectContainerInterface interface

Behavior type	Description
Behavior prior to WebSphere Process Server version 6.2	When you set and then retrieved a BusinessObjectInterface object from the CxObjectContainerInterface interface, you could use the Java equivalence operator “==” because the BusinessObjectInterface container that was retrieved was the same BusinessObjectInterface container that was set.
Behavior after WebSphere Process Server version 6.2	You must use the BusinessObjectInterface.isEquivalent(BOI) method.

Table 8. Behavior changes: Using the Java equivalence operator with the CxObjectContainerInterface interface (continued)

Behavior type	Description

Example of new behavior	The following JUnit test code demonstrates the old and new behavior.
	`CxObjectContainerInterface testCxObjectContainerInt;`
	`BusinessObjectInterface mB01, mB02, mB03;`
	`testCxObjectContainerInt.insertBusinessObject(mB01);`
	`testCxObjectContainerInt.setBusinessObject(1, mB01);`
	`BusinessObjectInterface mB02 = testCxObjectContainerInt.getBusinessObject(0));`
	`BusinessObjectInterface mB03 = testCxObjectContainerInt.getBusinessObject(1));`
	`assertTrue(mB01 == mB02);`
	`assertTrue(mB01 == mB03);`
	`assertTrue(mB02 == mB03);`
	This Java equivalence operator no longer works because the BusinessObjectInterface object that is returned by CxObjectContainerInterface.getBusinessObject(int index) is not the same Java object that was set to CxObjectContainerInterface.
	In the following code, the equivalence operator is replaced by the method BusinessObjectInterface.isEquivalent(BOI):
	`boolean result1 = mB01.isEquivalent(mB02)`
	`assertTrue(result1);`
	`boolean result2 = mB01.isEquivalent(mB03)`
	`assertFalse(result2);`
	`boolean result3 = mB02.isEquivalent(mB03)`
	`assertFalse(result3);`
	The cloned objects do not share a reference, and changes to the original BusinessObjectInterface object are not reflected in the cloned BusinessObjectInterface object:
	`hw1BOI.setAttrValue("Message", "hw1BOI_message changed");`
	`boolean result = mB01.isEquivalent(mB02;`
	`assertTrue(result);`
	`boolean result = mB01.isEquivalent(mB02;`
	`assertFalse(result);`
	`boolean result = mB02.isEquivalent(hw3BOI);`
	`assertFalse(result);`

Using weakly typed attribute data types for the BusObj class validData methods

The following table describes the previous behavior, current behavior, and an example of what to change if you previously had used weakly typed attribute data types in WebSphere InterChange Server or WebSphere Business Integration Server Express for the BusObj class validData methods.

Table 9. Behavior changes: Using weakly typed attribute data types for BusObj class validData methods

Behavior type	Description
Behavior prior to WebSphere Process Server version 6.2	For the BusObj class validData methods, attribute data types were weakly typed in WebSphere InterChange Server or WebSphere Business Integration Server Express. This allowed some odd data-type combinations. For example, if a business object had an attribute that was of type boolean but you used a set method that had a string parameter, you were able to set the string “not a boolean” into an attribute that was type boolean. As long as you used the getString method, they could get the string “not a boolean” back.

Table 9. Behavior changes: Using weakly typed attribute data types for BusObj class validData methods (continued)

Behavior type	Description
Behavior after WebSphere Process Server version 6.2	These attribute data types are now strongly typed. If a data type was valid but now is not valid, a CollaborationException exception is thrown, with message number 1802. Because WebSphere Process Server is strongly typed, you cannot put a String value into an Attribute of type boolean. Even if you used the Java conversions for strings into the boolean values of *true* and *false*, there is no way to return the original value of “not a boolean”. The only possible returned value is *true* or *false*. Therefore, attributes are now strongly typed for doubles-floats or int-long; these can be used interchangeably where Java provides automatic casting. However, as with any casting of types, you can expect some loss of precision when fields are demoted. If a type is not valid for the attribute to which it is set, but it used to be valid in WebSphere InterChange Server or WebSphere Business Integration Server Express, then a CollaborationException exception is thrown, with message number 1802. This is a new message number; the message definition is located in the InterchangeSystem.txt message file.

Example of new behavior	If a type is not valid for the Attribute to which it is set, but it used to be valid in WebSphere InterChange Server, a CollaborationException with message number 1802 will be thrown. This is a new message number, the Message definition is located in the InterchangeSystem.txt message file:
	`try`
	`{`
	`BusObj mBO = new BusObj("MasterBusinessObject");`
	`mBO.set("Attr_Two", "xxx");`
	`fail("Expected CollaborationException not thrown");`
	`}`
	`catch (CollaborationException e)`
	`{`
	`int a = e.getMsgNumber();`
	`String b = e.getSubType();`
	`String c = e.getMessage();`
	`String d = e.toString();`
	`assertEquals("exception_msgNumber", 1802, a);`
	`assertEquals("exception_type", "AttributeException", b);`
	`assertEquals("exception_message",`
	`"Error 1802 The attribute \"Attr_Two\" in SDO MasterBusinessObject is`
	`of type boolean and is not allowed to be set with a value \"xxx\" of`
	`type String. Error1802", c);`
	
	`assertEquals("exception_toString", "AttributeException: Error 1802 The`
	`attribute \"Attr_Two\" in SDO MasterBusinessObject is of type boolean`
	`and is not allowed to be set with a value \"xxx\" of type String.`
	`Error1802", d);`
	`}`

Migrating from WebSphere Studio Application Developer Integration Edition

To migrate 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 workspaces 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 6.2, version 6.1.x, and version 6.0.2.x. 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 6.2"
- "Deprecated features in WebSphere Process Server version 6.1.2" on page 206
- "Deprecated features in WebSphere Process server version 6.1" on page 206
- "Deprecated features in WebSphere Process Server version 6.0.2" on page 211
- "Deprecated features in WebSphere Process Server version 6.0.1" on page 213
- "Deprecated features in WebSphere Process Server version 6.0" on page 213

The following information summarizes what is deprecated, by version and release. Each section 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 6.2

BOCopy Service, Two methods: copyInto() and copyIntoShallow()

The following two methods within the BOCopy Service are deprecated: `copyInto()` and `copyIntoShallow()`.

Doing both a copy and a set at the same time will mask some of the problems you may have with the copy or the set. Because it is just as easy to copy and set them separately, the workaround is simple. Use copy() then set() rather than using a combined version of the API.

Recommended migration action:

Use the following methods instead of `copyInto()` and `copyIntoShallow()`:

- Instead of `copyInto()`, use: `copy()` then `set()`
- Instead of `copyIntoShallow()`, use `copyShallow()` then `set()`

CEI parameters used for stand-alone profile which are common with the CommonDB

Most of the CEI parameters used for stand-alone profiles which are common with the CommonDB are deprecated.

Recommended migration action:

If you are using a manageprofiles command in version 6.1.x or 6.0.2.x and are planning to use the same command in version 6.2, it has to be modified to use the new set of parameters.

Note: If you are using Profile Management Tool, then the GUI (graphical user interface) takes care of passing the correct parameters.
The following table describes the CEI parameters that are being deprecated. Beginning with version 6.2, use the same parameters as those for CommonDB for the corresponding CEI parameters. Examples of how to modify the CEI parameters appear below the table.

Table 10. Deprecated CEI parameters

CEI variable name	CommonDB variable name	Applicable database
nodeName	nodeName	All
ceiServerName	serverName	All
ceiDbExecuteScripts	dbDelayConfig	All
ceiJdbcClassPath	dbJDBCClasspath	All
ceiDbHostName	dbHostName	All
ceiDbPort	dbServerPort	All
ceiDbUser	dbUserId	All except MSSQL
ceiDbPassword	dbPassword	All except MSSQL
ceiOutputScriptDir	dbOutputscriptDir	All
ceiStorageGroup	dbStorageGroup	DB2 z/OS
ceiDbAliasName	dbSchemaName	DB2 z/OS
ceiDbSubSystemName	dbConnectionLocation	DB2 z/OS
ceiNativeJdbcClassPath	dbJDBCClasspath	DB2 iSeries® Native
ceiCollection	dbSchemaName	DB2 iSeries Native
ceiToolboxJdbcClassPath	dbJDBCClasspath	DB2 iSeries Toolbox
ceiCollection	dbSchemaName	DB2 iSeries Toolbox
ceiDbInformixDir	dbLocation	Informix
ceiDbServerName	dbInstance	Informix
ceiDbSysUser	dbSysUserId	Oracle
ceiDbSysPassword	dbSysPassword	Oracle

Example 1: manageprofiles

Here is an example which shows the old command and the new command if you are using the manageprofiles command. You no longer need to pass the CEI parameters except the ″ceiDBName″.

OLD

```
612 manageprofiles.bat -create -profileName -templatePath
\profileTemplates\default.wbiserver -dbType DB2_Universal -dbDelayConfig
false -dbCreateNew true -dbJDBCClasspath <classpath> -dbHostName localhost
-dbServerPort <port> -dbUserId <userid> -dbPassword <password>
```

```
-ceiDbProduct CEI_DB_DB2 -ceiDbExecuteScripts true -ceiJdbcClassPath
<classpath> -ceiDbHostName localhost -ceiDbPort <port> -ceiDbUser <userid>
-ceiDbPassword <password>
```

NEW

```
62 manageprofiles.bat -create -profileName -templatePath
\profileTemplates\default.wbiserver -dbDelayConfig false -dbType
DB2_Universal -dbJDBCClasspath <classpath> -dbHostName -dbServerPort <port>
-dbUserId <userid> -dbPassword <password>
```

″Everyone″ value used to map J2EE roles BPEAPIUser and TaskAPIUser

The possible use of the value ″Everyone″ to map J2EE roles BPEAPIUser and TaskAPIUserd is deprecated.

Recommended migration action:

If you used the value ″Everyone″ to map J2EE roles BPEAPIUser and TaskAPIUser, fix your Business Process Choreographer client applications by logging in before using the Business Process Choreographer APIs.

FailedEventManagerMBean interface and API

The following FailedEventManagerMBean interface, methods, and operations are deprecated:

- `com.ibm.wbiserver.manualrecovery.FailedEventWithParameters` (whole class)
- `com.ibm.wbiserver.manualrecovery.FailedEventManager` (methods)
- `FailedEventManagerMBean.xml` (operations)

Recommended migration action:

It is recommended that you switch to the new interface and MBean operations only if custom code is used to manage failed events with FailedEventManagerMBean. The suggested new interfaces, methods, and operations are shown in the following table.

Table 11. New interfaces, methods, and operations forFailedEventManagerMBean

Deprecated interface, operation, or method	New interface, operation, or method
com.ibm.wbiserver.manualrecovery. FailedEventWithParameters	com.ibm.wbiserver.manualrecovery.SCAEvent
com.ibm.wbiserver.manualrecovery. FailedEventManager	
• List getFailedEventsForDestination(String destModuleName, String destComponentName, String destMethodName, int pagesize) throws FailedEventReadException; • List getFailedEventsForTimePeriod(Date begin, Date end, int pagesize) throws FailedEventReadException;	List<FailedEvent> queryFailedEvents(QueryFilters queryFilters, int offset, int maxRows) throws FailedEventReadException;
FailedEventWithParameters getFailedEventWithParameters(String msgId) throws FailedEventDataException;	SCAEvent getEventDetailForSCA(FailedEvent failedEvent) throws FailedEventDataException;

Table 11. New interfaces, methods, and operations forFailedEventManagerMBean (continued)

Deprecated interface, operation, or method	**New interface, operation, or method**
void discardFailedEvents(String[] msgIds) throws DiscardFailedException;	void discardFailedEvents(List<FailedEvent> failedEvents) throws DiscardFailedException;
void resubmitFailedEvents(String[] msgIds) throws ResubmissionFailedException;	void resubmitFailedEvents(List failedEvents) throws ResubmissionFailedException;
FailedEventManagerMBean.xml	
• getFailedEventsForDestination • getFailedEventsForTimePeriod	queryFailedEvents
getFailedEventWithParameters	getEventDetailForSCA
discardFailedEvents	discardFailedEvents with the following parameters: • name=″failedEvents″ • description=″A list of failed events″ • type=″java.util.List″
resubmitFailedEvents	resubmitFailedEvents • name=″failedEvents″ • description=″A list of failed events″ • type=″java.util.List″

WebSphere Connect JDBC Drivers (from DataDirect) for Microsoft SQL Server

The WebSphere Connect JDBC Drivers (from DataDirect) for Microsoft SQL Server that are shipped on the WebSphere Application Server Supplemental CDs are no longer shipped on the WebSphere Application Server, version 7, CDs.

Recommended migration action:

You must migrate any Microsoft SQL databases that are using the shipped DataDirect drivers to instead use the new JDBC driver that is provided by Microsoft. The new JDBC driver is not currently supported by WebSphere Process Server and WebSphere Enterprise Service Bus, but it will be supported in the future. You can either change to another database type (such as the Microsoft SQL embedded driver), or you can wait for the new JDBC driver to be supported by WebSphere Process Server and WebSphere Enterprise Service Bus, and migrate at that time.

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)

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:

- com.ibm.websphere.rsadapter.WSProceduralPushDownHelper
- com.ibm.websphere.rsadapter.WSPushDownHelper

- com.ibm.websphere.rsadapter.WSPushDownHelperFactory
- com.ibm.websphere.rsadapter.WSRelationalPushDownHelper

Recommended migration action:

Convert CMP Entity Beans to use a relational data source, or have the CMP entity bean replaced by a different supported data persistence model.

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.

JACL scripts (deprecated in WebSphere Application Server version 6.1)

JACL script files are deprecated in WebSphere Process Server to maintain consistency with the deprecation of JACL scripts in WebSphere Application Server.

Recommended migration action:

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:

1. <*install_root*>\ProcessChoreographer\admin\bpcTemplates.jacl
2. <*install_root*>\ProcessChoreographer\config\bpeconfig.jacl
3. <*install_root*>\ProcessChoreographer\config\bpeunconfig.jacl
4. <*install_root*>\ProcessChoreographer\config\bpeupgrade.jacl
5. <*install_root*>\ProcessChoreographer\config\clientconfig.jacl

IBM Web Services Client for C++

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.

Recommended migration action:

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.

Business Process Choreographer

Generic Business Process EJB API

- The getAutoDelete() function from ProcessTemplateData is deprecated.

  Recommended migration action:

  Use method getAutoDeletionMode() to query how auto deletion is handled for the corresponding process template.
- The exception SpecificFaultReplyException is deprecated.

Recommended migration action:

No action is required. This exception is only needed to handle WSIF messages, which are no longer supported.

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:

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

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

The following Business Object methods are deprecated:

- com.ibm.websphere.bo.BOFactory.createByClass(java.lang.Class iterfaceClass);
- com.ibm.websphere.bo.BOType.getTypeByClass(java.lang.Class className);

Recommended migration action:

These methods will raise ″function not supported″ exceptions if they are called in version 6.1.

Common Event Infrastructure

Creation and editing of user-visible Common Base Events are deprecated.

Recommended migration action:

You can now use the tools to specify the Business Object data that is to be included in monitored emitted events.

zOS

The requirement to bind a String object into JNDI at esb/messageLogger/qualifier is deprecated.

Recommended migration action:

The Message Logger primitives will now store message information within the CommonDB database. Where necessary, during the profile augmentation phase, a WebSphere variable called ESB_MESSAGE_LOGGER_QUALIFIER will now be created and its value set to that of the chosen CommonDB schema qualifier.

WebSphere InterChange Server

The APIs (application programming interfaces) listed in Supported WebSphere InterChange Server APIs are no longer deprecated.

Note: These APIs were formerly deprecated in WebSphere Process Server version 6.0.2.

Recommended migration action:

You should use these APIs only for applications with migrated WebSphere InterChange Server components. In all other cases, you should use the Service Data Objects for WebSphere Process Server.

WebSphere Enterprise Service Bus (WESB)

The current method to identify an SSL repertoire to be used when WESB communicates with a secured WSRR instance has been deprecated.

Recommended migration action:

A new property has been added to WSRR definitions to allow the specification of such a repertoire.

Deprecated features in WebSphere Process Server version 6.0.2

Human Task Manager

The task context variable %htm:task.clientDetailURL% is no longer required, and thus has been deprecated.

Recommended migration action:

No action is required.

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.

Recommended migration action:

Use the customizeable e-mail feature for escalations.

The following Task object methods that were deprecated in version 6.0 are no longer deprecated:

getInputMessageTypeName()
getOutputMessageTypeName()

Recommended migration action:

You can now use these methods.

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)

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

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

WebSphere InterChange Server

Note: These APIs are no longer deprecated in version 6.1.

The APIs (application programming interfaces) listed in Supported WebSphere InterChange Server APIs are deprecated.

Recommended migration action:

Code written for WebSphere Process Server should not use these interfaces.

IBM WebSphere InterChange Server Access for Enterprise JavaBeans (EJB) support is deprecated.

Recommended migration action:

Applications developed for use with WebSphere Process Server should not use Access for Enterprise JavaBeans.

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:

  getInputMessageTypeTypeSystemName()

  getOutputMessageTypeTypeSystemName()
- The following methods of the ProcessInstanceData object are no longer supported:

  getInputMessageTypeTypeSystemName()

  getOutputMessageTypeTypeSystemName()
- The following methods of the ActivityInstanceData object are no longer supported:

  getInputMessageTypeTypeSystemName()

  getOutputMessageTypeTypeSystemName()
- The following methods of the ActivityServiceTemplateData object are no longer supported:

  getInputMessageTypeTypeSystemName()

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:

  getInputMessageTypeName()

  getOutputMessageTypeName()

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.

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

The following sections describe specific errors and exceptions that may occur in a version-to-version migration and provide steps you can follow to understand and resolve these problems.

- "Application installation error" on page 141
- "Application server error" on page 141
- "Exceptions: database connectivity, loading, or missing class" on page 142
- "Out of memory error" on page 142
- "Profile creation error" on page 143
- "Profile migration error" on page 143
- "Servlet error" on page 144
- "Synchronization error" on page 144

Application installation error

If you select the option for the migration process to install the enterprise applications that exist in the version 6.1.x or 6.0.2.x configuration into the new version 6.2 configuration, you might encounter some error messages during the application-installation phase of migration.

The applications that exist in the version 6.1.x or 6.0.2.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.

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

- Fix the problems in the version 6.1.x or 6.0.2.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 6.2 configuration using the administrative console or an install script.

Application server error

After you migrate a managed node to version 6.2, 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.

Exceptions: database connectivity, loading, or missing class

Never change any WebSphere Application Server variables that are configured as a part of profile creation.

If you modify these values incorrectly in old profile, you might get database connectivity, loading, or other missing class exceptions, such as:

```
10/25/08 13:22:39:650 GMT+08:00] 0000002e J2CUtilityCla E   J2CA0036E: An
exception occurred while invoking method setDataSourceProperties on
com.ibm.ws.rsadapter.spi.WSManagedConnectionFactoryImpl used by resource
jdbc/com.ibm.ws.sib/ewps6101.Messaging-BPC.cwfpcCell01.Bus :
com.ibm.ws.exception.WsException: DSRA0023E: The DataSource implementation
class "com.ibm.db2.jcc.DB2XADataSource" could not be found.DB2,
```

Derby, and SQL Embedded JDBC drivers are bundled with the WebSphere Process Server product installation. If you need to change these drivers to any higher version, you must copy drivers on same location where they exists in the product installation, as follows:

- **Derby**: %*was.install.root*%\derby\lib
- **DB2**: %*was.install.root*%/universalDriver_wbi/lib

- **SQL**: %*was.install.root*%lib

If you need a new JDBC provider and datasource for your application, you can create these resources by selecting a valid jdbcclasspath and setting the WebSphere Application Server variable accordingly. For example, if you need DB2 at cell level which doesn't exist earlier in your installation, you could use the following procedure.

1. In the administrative console, navigate to: **Resources** → **JDBC** → **JDBC Providers** → **DB2 Universal JDBC Driver Provider (XA)**.
2. In the **Class path** box, set the following paths:
   - DB2UNIVERSAL_JDBC_DRIVER_PATH =%was.install.root%/universalDriver_wbi/lib
   - DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH=""

   If you need your own drivers, set the following path: DB2UNIVERSAL_JDBC_DRIVER_PATH=%myDriverLocation%

Out of memory error

If either the WBIPreUpgrade or WBIPostUpgrade command-line utility fail due to Out of Memory problems, you can increase the heap size to a number that takes into consideration the size and scope of the environment being migrated, as well as what the machine will allow.

For instructions on how to increase the heap size, use the procedure described in Solution 4 of the following technote: Instructions for handling certain Out of Memory conditions.

Profile creation error

While you are using the version 6.2 migration wizard to create a profile when migrating a configuration, you might see the following profile-creation error messages.

```
profileName: profileName cannot be empty
profilePath: Insufficient disk space
```

These error messages might be displayed if you enter a profile name that contains an incorrect character such as a space. Rerun the migration wizard, and verify that there are no incorrect characters in the profile name such as a space, quotes, or any other special characters.

Profile migration error

When you use the migration wizard to migrate a profile from WebSphere Process Server version 6.1.x or 6.0.2.x to version 6.2 on a Solaris x64 processor-based system, the migration might fail during the WBIPostUpgrade step.

You might see messages similar to the following in *profile_root*/logs/WASPostUpgrade.*time_stamp*.log:

```
MIGR0327E: A failure occurred with stopNode.
MIGR0272E: The migration function cannot complete the command.
```

WebSphere Process Server version 6.1.x or 6.0.2.x uses a Java virtual machine (JVM) in 32-bit mode. The migration wizard for WebSphere Process Server version

6.2 calls the WBIPostUpgrade.sh script, which attempts to run the JVM for version 6.1.x or 6.0.2.x in the 64-bit mode when the server stops the version 6.1.x or 6.0.2.x node.

Complete the following actions to remove the incomplete profile and enable WebSphere Process Server to correctly migrate the version 6.1.x or 6.0.2.x profile:

1. On a command line, change to the *install_root*/bin directory.

   For example, type the following command:

   ```
   cd /opt/IBM/WebSphere/Procserver/bin
   ```
2. Locate the WBIPostUpgrade.sh script in the *install_root*/bin directory, and make a backup copy.
3. Open the WBIPostUpgrade.sh or WBIPostUpgrade.bat file in an editor, and perform the following actions:
   a. Locate the following line of code:

      UNIX Linux

      ```
      "$binDir" /setupCmdLine.sh
      ```

      Windows

      ```
      call "%~dp0setupCmdLine.bat" %*
      ```
   b. Insert the following line of code after the code that was identified in the previous step:

      ```
      JVM_EXTRA_CMD_ARGS=""
      ```
   c. Save the changes.
4. Repeat steps 2 through 4 with theWASPostUpgrade.sh or the WASPostUpgrade.bat file.
5. Delete the incomplete version 6.2 profile that was created during the migration process. Use the following procedure.
   a. Open a command prompt and run one of the following commands, based on your operating system:
      - i5/OS **On i5/OS platforms:** `manageprofiles -delete -profileName` *`profile_name`*
      - Linux UNIX **On Linux and UNIX platforms:** `manageprofiles.sh -delete -profileName` *`profile_name`*
      - Windows **On Windows platforms:** `manageprofiles.bat -delete -profileName` *`profile_name`*

      The variable *profile_name* represents the name of the profile that you want to delete.
   b. Confirm that the profile deletion has completed by checking the following log file:
      - i5/OS **On i5/OS platforms:** *user_data_root*/profileRegistry/logs/manageprofiles/*profile_name*_delete.log
      - Linux UNIX **On Linux and UNIX platforms:** *install_root*/logs/manageprofiles/*profile_name*_delete.log
      - Windows **On Windows platforms:** *install_root*\logs\manageprofiles\profile_name_delete.log
6. Delete the *profile_root* directory of the version 6.2 profile that was removed in the previous step.
7. Rerun the migration wizard.

Servlet error

In a network deployment environment, if the error `SRVE0026E: [Servlet Error]-[com/ibm/wbiservers/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.

Synchronization error

If synchronization fails when you migrate a managed node to version 6.2, the server might not start.

You might receive messages similar to the following when you migrate a managed node to version 6.2:

```
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/62AppServer/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 6.2 configuration level.
- The managed node that you are trying to migrate is at a version 6.2 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 .

Related concepts

Premigration considerations for Business Process Choreographer
If your servers run Business Process Choreographer, you should be aware of certain things that you need to plan and take into consideration before you migrate Business Process Choreographer.

Related tasks

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

Related reference

WBIPreUpgrade command-line utility
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-line utility
Use the WBIPostUpgrade command for WebSphere Process Server to retrieve the profile configuration that was saved by the WBIPreUpgrade command at the *backupDirectory* that you had specified.

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

Troubleshooting migration from WebSphere InterChange Server or WebSphere Business Integration Server Express

Find solutions to problems you encounter with migration as well as instructions for turning on logging and tracing.

Related concepts

Limitations when migrating from WebSphere InterChange Server or WebSphere Business Integration Server Express
Some characteristics of WebSphere InterChange Server or WebSphere Business Integration Server Express are not precisely duplicated by WebSphere Process Server. Therefore you might have to modify your applications after migration to get them to perform as they did in WebSphere InterChange Server or WebSphere Business Integration Server Express.

Related reference

Postmigration considerations
When applications have been migrated from WebSphere InterChange Server or WebSphere Business Integration Server Express to WebSphere Process Server, special attention is required in some areas to enable migrated applications to function in WebSphere Process Server consistently with their intended function due to differences between the architectures of WebSphere Process Server and WebSphere InterChange Server or WebSphere Business Integration Server Express.

Premigration considerations
Consider these guidelines for the development of integration artifacts for WebSphere InterChange Server or WebSphere Business Integration Server Express to ease the migration of WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts to WebSphere Process Server.

Enabling logging and tracing for supported WebSphere InterChange Server or WebSphere Business Integration Server Express APIs

Enable logging and tracing for the supported WebSphere InterChange Server or WebSphere Business Integration Server Express APIs through the administrative console.

About this task

If your migrated application includes any supported WebSphere InterChange Server or WebSphere Business Integration Server Express APIs, you can enable logging and tracing for them for troubleshooting purposes.

Procedure

1. Launch the administrative console.
2. From the left (navigation) panel, select **Troubleshooting > Logs and Trace**.
3. In the right panel, select the name of the server on which you want to enable logging and tracing.
4. In the right panel, under ″General properties,″ select **Change Log Level Details**.
5. Select the Runtime tab. (Selecting the Runtime tab allows you to make this change in real time without requiring you to restart the server.)
6. Add the name of the package followed by `=all` to the list of logged packages in the box on the screen. Separate this new entry from any existing entries with a colon. For example, `CxCommon=all`. In this case, `CxCommon` is the name of the package for a set of supported WebSphere InterChange Server or WebSphere Business Integration Server Express APIs. Specifying `all` enables all logging and tracing. See Supported WebSphere InterChange Server APIs for a list of the APIs, including their package names.
7. Select **Apply**.

8. To keep this configuration after the server is restarted, select the **Save runtime changes to configuration as well** check box.
9. Select **OK**.
10. When the next screen appears, select **Save** to save your changes.

Related information

Supported WebSphere InterChange Server APIs

Failure trying to serialize an object that is not serializable in a migrated BPEL file

If a serialization failure occurs in a BPEL file generated by the migration, you might be able to modify it to prevent the failure from occurring.

Problem: A serialization failure occurs in a custom snippet node of a business process execution language (BPEL) file generated by migration because an attempt is made to serialize an object that is not serializable.

Cause: In WebSphere InterChange Server or WebSphere Business Integration Server Express, a Collaboration Template is compiled into a single Java class. In WebSphere Process Server, each node in a BPEL file might compile into a separate Java class. In WebSphere InterChange Server or WebSphere Business Integration Server Express, a variable can be declared once and shared throughout the various steps of a Collaboration Template. To simulate that behavior in the migrated BPEL file, each variable used in a code snippet must be retrieved at the start of the snippet and saved at the end of the snippet. Variables defined in WebSphere InterChange Server or WebSphere Business Integration Server Express Port definitions become BPEL variables. These are retrieved into BusObj variables at the beginning of each snippet (if referenced in the snippet) and saved back to the BPEL variables at the end of each snippet. For example, a retrieval at the beginning of snippets looks like this:

```
BusObj tempBusObj = null;if (tempBusObj_var != null) { tempBusObj =
      new BusObj(tempBusObj_var); };
```

and a save at the end of snippets looks like this:

```
if (tempBusObj == null) { tempBusObj_var = null; } else { tempBusObj_var =
      tempBusObj.getBusinessGraph(); }
```

Other variables used in the WebSphere InterChange Server or WebSphere Business Integration Server Express snippet code are serialized and stored as a String in a BPEL variable named *CollabTemplateName_var*. These variables are deserialized at the beginning of each BPEL snippet, and then serialized and saved at the end of each BPEL Snippet that they are referenced in. For example, objects are retrieved like this:

```
BusObj tempBusObj = (BusObj)BaseCollaboration.deserialize
  (FrontEndCollab_var.getString("tempBusObj"));
```

and objects are saved like this:

```
FrontEndCollab_var.setString("tempBusObj", BaseCollaboration.serialize(tempBusObj));
```

If the type of the object being serialized is not serializable, then using serialize and deserialize will fail when the BPEL is run.

Solution: After migration, modify the BPEL file as follows:

- For any variable that is not Java-serializable, update the BPEL snippets to remove the serialization and deserialization statements. If the variable needs to be shared across snippets (instead of being re-created in each snippet) another method must be used to preserve the value of the variable across snippets.
- Manually define BPEL variables for variables of type BusObj that are not declared in the WebSphere InterChange Server or WebSphere Business Integration Server Express Port definitions but are used on Partner Invokes. This is a manual step because variables used during invokes in WebSphere Process Server must be strongly typed, and the migration tools cannot accurately determine that type from the WebSphere InterChange Server or WebSphere Business Integration Server Express snippets.

  Note: The naming convention used by the migration tools is to add `_var` to the name of the variable in the snippet code when naming the BPEL variables. For example, for a variable called tempBusObj in the snippet code, the migration tools will create a BPEL variable named `tempBusObj_var`.
- For variables that must be declared manually as BPEL variables, change the BPEL snippet code so that it uses the ″deserialize/serialize″ method of preserving these variables rather than the ″retrieve from/store into BPEL variable″ method of preserving these variables.

New behavior for heritage APIs in WebSphere Process Server, version 6.2

In version 6.2 of WebSphere Process Server, heritage APIs use WebSphere Process Server Service Data Objects to store attribute states and data that were formerly stored by the BusinessObjectInterface interface. As a result, the behavior of some method calls in the BusinessObjectInterface and CxObjectContainerInterface interfaces have changed.

The major change to the heritage APIs (HAPIs) in WebSphere Process Server version 6.2 is that the WebSphere InterChange Server BusinessObjectInterface interface is no longer the root storage object for HAPI. Instead, a WebSphere Process Server Service Data Object (SDO) is now used to store attribute states and data.

If you use the Java equivalence operator and weakly typed attribute principles, the behavior of the method calls in the BusinessObjectInterface and CxObjectContainerInterface interfaces is different, as described in the following sections:

- “Using the Java equivalence operator when performing a Set operation followed by a Get operation” on page 197
- “Using the Java equivalence operator when setting a BusinessObjectInterface object to more than one target attribute” on page 198
- “Using the Java equivalence operator when you set and retrieve a BusinessObjectInterface object from the CxObjectContainerInterface interface” on page 199
- “Using weakly typed attribute data types for the BusObj class validData methods” on page 200

Using the Java equivalence operator when performing a Set operation followed by a Get operation

A different BusinessObjectInterface object is returned when performing a Set operation followed by a Get operation of a BusinessObjectInterface object for one target attribute. The following table describes the previous behavior, current behavior, and an example of what to change if you previously had used the Java equivalence operator when performing a Set operations followed by a Get operation.

Table 12. Behavior changes: Using the Java equivalence operator with Set and Get operations

Behavior type	Description
Behavior prior to WebSphere Process Server version 6.2	The same BusinessObjectInterface container that was set was also retrieved, and you could use the Java equivalence operator “==” to determine if they were the same. Example: `boolean b = (JavaObjectA == JavaObjectB)`
Behavior after WebSphere Process Server version 6.2	The original BusinessObjectInterface container is discarded, and when you perform a Get operation to retrieve the BusinessObjectInterface object, a new container is created. The returned container is not the same object, but the root object that it wraps is the same object. A new method, isEquivalent, has been added to the BusinessObjectInterface class: BusinessObjectInterface.isEquivalent(BOI). When you want to determine if the two BusinessObjectInterface objects are equivalent, use the isEquivalent method to perform the comparison.

Example of new behavior	The following example shows the use of isEquivalent. You have a BusinessObjectInterface object of type MasterBusinessObject with the attribute Attr_Nine, which is a BusinessObjectInterface object of type HelloWorld:
	`BusinessObjectInterface mboBOI, hw1BOI, hw2BOI;`
	`hw1BOI.setAttrValue("Message", "hw1BOI_message");`
	`hw1BOI.setVerb("Create");`
	`mboBOI.setAttrValue("Attr_Nine", hw1BOI);`
	`hw2BOI = mboBOI.getAttrValue("Attr_Nine");`
	Instead of:
	`boolean result = (hw1BOI == hw2BOI); assertTrue(result);`
	Use this:
	`boolean result = hw1BOI.isEquivalent(hw2BOI); assertTrue(result);`

Using the Java equivalence operator when setting a BusinessObjectInterface object to more than one target attribute

Setting a BusinessObjectInterface object to more than one target attribute sets a cloned object. This applies both to elements of a BusObjArray class and to multiple target attributes. The following table describes the previous behavior, current behavior, and an example of what to change if you previously had used the Java equivalence operator when setting a BusinessObjectInterface object to more than one target attribute.

Table 13. Behavior changes: Using the Java equivalence operator with more than one target attribute

Behavior type	Description
Behavior prior to WebSphere Process Server version 6.2	You could set a BusinessObjectInterface object to multiple locations, and all of the locations contained a reference to the original BusinessObjectInterface object. If you changed the attribute in one BusinessObjectInterface object, that change was reflected in all of the other references for that object.
Behavior after WebSphere Process Server version 6.2	Service Data Object (SDO) rules prevent you from setting the same SDO to more than one target property. If you try to set the SDO to more than one target property, the SDO moves from one attribute to the next, leaving a ″null″ value at the previous attribute location. Now, instead of leaving a ″null″ value when the BusinessObjectInterface object is set to a second, third, and so on, location, the object is cloned into the multiple locations. For example, you have a BusinessObjectInterface object of the type MasterBusinessObject, with the attributes Attr_Nine and Attr_Eleven that are of the type HelloWorld. If you set the same HelloWorld object to both attributes, then Attr_Nine is assigned the original object, and Attr_Eleven is assigned a clone. The clone is a snapshot of the object at the time that it is cloned. If you want to determine if two BusinessObjectInterface objects are equivalent, do not use the Java equivalence operator; instead, use the isEquivalent method to perform the comparison.

Example of new behavior	The following example shows the use of isEquivalent and clones. You have a BusinessObjectInterface object of type MasterBusinessObject, with the attributes Attr_Nine and Attr_Eleven that are of type HelloWorld:
	`BusinessObjectInterface mboBOI;`
	`BusinessObjectInterface hw1BOI, hw2BOI, hw3BOI;`
	`hw1BOI.setAttrValue("Message", "hw1BOI_message");`
	`hw1BOI.setVerb("Create");`
	`mboBOI.setAttrValue("Attr_Nine", hw1BOI);`
	`mboBOI.setAttrValue("Attr_Eleven", hw1BOI);`
	`hw2BOI = mboBOI.getAttrValue("Attr_Nine");`
	`hw3BOI = mboBOI.getAttrValue("Attr_ Eleven ");`
	Instead of:
	`boolean result = hw2BOI == hw3BOI); assertTrue(result);`
	Use isEquivalent instead:
	`boolean result = hw2BOI.isEquivalent(hw3BOI); assertTrue(result);`
	The cloned objects do not share a reference, and changes to the original BusinessObjectInterface object are not reflected in the cloned BusinessObjectInterface object:
	`hw1BOI.setAttrValue("Message", "hw1BOI_message changed");`
	`boolean result = hw1BOI.isEquivalent(hw2BOI);`
	`assertTrue(result);`
	`boolean result = hw1BOI.isEquivalent(hw3BOI);`
	`assertFalse(result);`
	`boolean result = hw2BOI.isEquivalent(hw3BOI);`
	`assertFalse(result);`

Using the Java equivalence operator when you set and retrieve a BusinessObjectInterface object from the CxObjectContainerInterface interface

The following table describes the previous behavior, current behavior, and an example of what to change if you previously had used the Java equivalence operator when setting and retrieving a BusinessObjectInterface object from the CxObjectContainerInterface interface.

Table 14. Behavior changes: Using the Java equivalence operator with the CxObjectContainerInterface interface

Behavior type	Description
Behavior prior to WebSphere Process Server version 6.2	When you set and then retrieved a BusinessObjectInterface object from the CxObjectContainerInterface interface, you could use the Java equivalence operator "==" because the BusinessObjectInterface container that was retrieved was the same BusinessObjectInterface container that was set.
Behavior after WebSphere Process Server version 6.2	You must use the BusinessObjectInterface.isEquivalent(BOI) method.

Example of new behavior	The following JUnit test code demonstrates the old and new behavior.
	`CxObjectContainerInterface testCxObjectContainerInt;`
	`BusinessObjectInterface mB01, mB02, mB03;`
	
	`testCxObjectContainerInt.insertBusinessObject(mB01);`
	`testCxObjectContainerInt.setBusinessObject(1, mB01);`
	`BusinessObjectInterface mB02 = testCxObjectContainerInt.getBusinessObject(0));`
	`BusinessObjectInterface mB03 = testCxObjectContainerInt.getBusinessObject(1));`
	`assertTrue(mB01 == mB02);`
	`assertTrue(mB01 == mB03);`
	`assertTrue(mB02 == mB03);`
	
	This Java equivalence operator no longer works because the BusinessObjectInterface object that is returned by CxObjectContainerInterface.getBusinessObject(int index) is not the same Java object that was set to CxObjectContainerInterface.
	
	In the following code, the equivalence operator is replaced by the method BusinessObjectInterface.isEquivalent(BOI):
	`boolean result1 = mB01.isEquivalent(mB02)`
	`assertTrue(result1);`
	`boolean result2 = mB01.isEquivalent(mB03)`
	`assertFalse(result2);`
	`boolean result3 = mB02.isEquivalent(mB03)`
	`assertFalse(result3);`
	
	The cloned objects do not share a reference, and changes to the original BusinessObjectInterface object are not reflected in the cloned BusinessObjectInterface object:
	`hw1BOI.setAttrValue("Message", "hw1BOI_message changed");`
	`boolean result = mB01.isEquivalent(mB02;`
	`assertTrue(result);`
	`boolean result = mB01.isEquivalent(mB02;`
	`assertFalse(result);`
	`boolean result = mB02.isEquivalent(hw3BOI);`
	`assertFalse(result);`

Using weakly typed attribute data types for the BusObj class validData methods

The following table describes the previous behavior, current behavior, and an example of what to change if you previously had used weakly typed attribute data types in WebSphere InterChange Server or WebSphere Business Integration Server Express for the BusObj class validData methods.

Table 15. Behavior changes: Using weakly typed attribute data types for BusObj class validData methods

Behavior type	Description
Behavior prior to WebSphere Process Server version 6.2	For the BusObj class validData methods, attribute data types were weakly typed in WebSphere InterChange Server or WebSphere Business Integration Server Express. This allowed some odd data-type combinations. For example, if a business object had an attribute that was of type boolean but you used a set method that had a string parameter, you were able to set the string “not a boolean” into an attribute that was type boolean. As long as you used the getString method, they could get the string “not a boolean” back.
Behavior after WebSphere Process Server version 6.2	These attribute data types are now strongly typed. If a data type was valid but now is not valid, a CollaborationException exception is thrown, with message number 1802. Because WebSphere Process Server is strongly typed, you cannot put a String value into an Attribute of type boolean. Even if you used the Java conversions for strings into the boolean values of *true* and *false*, there is no way to return the original value of “not a boolean”. The only possible returned value is *true* or *false*. Therefore, attributes are now strongly typed for doubles-floats or int-long; these can be used interchangeably where Java provides automatic casting. However, as with any casting of types, you can expect some loss of precision when fields are demoted. If a type is not valid for the attribute to which it is set, but it used to be valid in WebSphere InterChange Server or WebSphere Business Integration Server Express, then a CollaborationException exception is thrown, with message number 1802. This is a new message number; the message definition is located in the InterchangeSystem.txt message file.

Table 15. Behavior changes: Using weakly typed attribute data types for BusObj class validData methods (continued)

Behavior type	Description

Example of new behavior	If a type is not valid for the Attribute to which it is set, but it used to be valid in WebSphere InterChange Server, a CollaborationException with message number 1802 will be thrown. This is a new message number, the Message definition is located in the InterchangeSystem.txt message file:
	`try`
	`{`
	`BusObj mBO = new BusObj("MasterBusinessObject");`
	`mBO.set("Attr_Two", "xxx");`
	`fail("Expected CollaborationException not thrown");`
	`}`
	`catch (CollaborationException e)`
	`{`
	`int a = e.getMsgNumber();`
	`String b = e.getSubType();`
	`String c = e.getMessage();`
	`String d = e.toString();`
	`assertEquals("exception_msgNumber", 1802, a);`
	`assertEquals("exception_type", "AttributeException", b);`
	`assertEquals("exception_message",`
	`"Error 1802 The attribute \"Attr_Two\" in SDO MasterBusinessObject is`
	`of type boolean and is not allowed to be set with a value \"xxx\" of`
	`type String. Error1802", c);`
	
	`assertEquals("exception_toString", "AttributeException: Error 1802 The`
	`attribute \"Attr_Two\" in SDO MasterBusinessObject is of type boolean`
	`and is not allowed to be set with a value \"xxx\" of type String.`
	`Error1802", d);`
	`}`

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 character set (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
1001 Hillsdale Blvd., Suite 400
Foster City, CA 94404
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 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 (^{R} or ^{TM}), 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.

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

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

Java and JavaBeans are trademarks of Sun Microsystems, Inc. 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 Multiplatforms, Version 6.2

Printed in USA