WebSphere Process Server for z/OS

Migrating WebSphere Process Server

Version 7.0.0

30 April 2010

This edition applies to version 7, release 0, modification 0 of WebSphere Process Server for z/OS (product number 5655-N53) and to all subsequent releases and modifications until otherwise indicated in new editions.

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

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

Contents

Migrating: Version-to-version

With version-to-version migration, applications can be migrated to the new version of the product by redeploying them, by updating the applications using the authoring tools and redeploying them, or by using the runtime migration tools which preserve all the configuration information and automatically redeploy the applications.

Migration overview

The process of moving applications, configuration, and databases from a prior version of WebSphere® Process Server to a more recent version of WebSphere Process Server is referred to as version-to-version migration, or simply migration.

What is version-to-version migration?

Version-to-version migration refers to the movement of profiles, applications and data associated with an earlier version of WebSphere Process Server to a newly installed version of WebSphere Process Server.

Version-to-version migration overview

Version-to-version migration, or simply migration, refers to the process of moving applications that have been developed on prior releases of WebSphere Process Server to version 7.0. Migration can be accomplished using a set of migration facilities provided by WebSphere Integration Developer or WebSphere Business Modeler for migrating applications, or migration can be accomplished in a production environment using a set of runtime migration procedures and tools for migrating the entire production configuration, applications, and databases.

In WebSphere Integration Developer and WebSphere Business Modeler, applications and workspaces developed using earlier versions can be imported and migrated to version 7.0. Once the applications have been migrated to version 7.0, they can either be directly deployed to a version 7.0 runtime, or they can be enhanced to exploit new capability delivered in version 7.0 and then be deployed. This style of migration is referred to as artifact migration.

The migration of applications deployed to production environments goes well beyond the scope of redeployment of the applications to the new version. The configuration of the production topology, the product databases, and the product data in the databases are all migrated to version 7.0 by a consistent set of BPM procedures and tools. The process associated with the set of procedures and tools for migrating production configuration, applications, and databases is referred to as runtime migration.

The BPM products leveraging the common runtime migration procedures and tools include:

- WebSphere Dynamic Process Edition
- WebSphere Business Services Fabric
- WebSphere Process Server
- WebSphere Enterprise Service Bus
- WebSphere Business Monitor

- WebSphere Business Compass

Applications can also be manually redeployed from a production environment that is the source of the migration to a parallel target production environment. This style of migration is referred to as manual migration.

Migrating multiple products

The runtime migration method provides support for migrating multiple BPM products that are installed and configured together in the source environment. For example, if the source installation directory of the migration contains WebSphere Process Server and WebSphere Business Monitor and a set of profiles that have been augmented by both products, the runtime migration method provides support for migrating that source environment to a target environment that contains both products installed into the same target installation directory.

Product updates

The version-to-version migration process differs from the process of applying interim fixes and updates to the development and production environments. For information on updates in the forms of interim fixes, fix packs, and refresh packs, see the "updating" topic for your BPM product.

Migrating heritage products

The version-to-version migration process differs from the process of migrating heritage products to WebSphere Process Server. For more information about migrating heritage products, see “Migrating: Heritage products” on page 91.

BPM Migration roadmap

The WebSphere Process Server migration roadmap shows the high-level tasks involved in a version-to-version migration.

Use the following flow diagram and high-level migration task descriptions to learn about the tasks involved in a version-to-version migration.

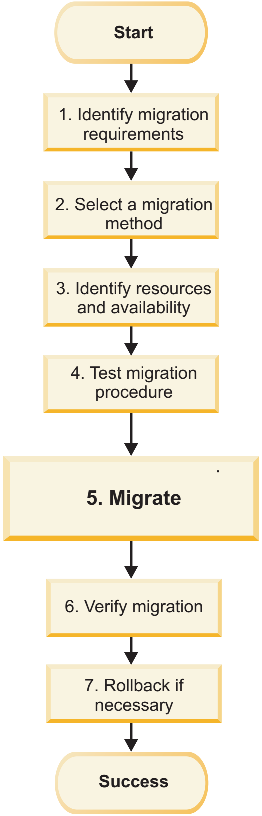
Start
1. Identify migration requirements
2. Select a migration method
3. Identify resources and availability
4. Test migration procedure
5. Migrate
6. Verify migration
7. Rollback if necessary
Success

Figure 1. WebSphere Process Server migration roadmap for version-to-version migration

1. Identify the migration requirements

Identifying the migration requirements is the first step in planning your migration.

See the Migration method comparison topic for a list of the set of considerations involved in the migration process.

If the goal for the migration is to exploit new capabilities delivered in Version 7.0 review the new features of WebSphere Process Server version 7.0 described in What is new in this release.

2. Select a migration method

There are three migration methods to choose from when migrating:

- Runtime migration
- Manual migration
- Artifact migration

To review the migration methods and determine which migration method fits your requirements see the Migration methods topic.

3. Identify resources and availability

When planning your migration, it is critical to identify the availability of all of the resources you need for the migration, including:

- Human resources: How many people and what skill level is needed? What is the timeframe for the human resource need?
- Hardware and software resources: What hardware or software do you need to acquire to ensure a successful migration?

To learn more about hardware and software requirements for WebSphere Process Server version 7.0, see Hardware and software requirements.

4. Test the migration procedure

Before performing the migration, thoroughly test the migration procedure:

- Test your applications in a new environment.
- Test your migration procedure in a staging environment.
- Practice your rollback plan on a test system.

When planning which test or tests will best suit your migration, keep in mind the necessary resources to make the test successful.

5. Migrate

Use the migration procedures associated with the migration method you choose to migrate your environment.

6. Verify migration

After performing the migration, use one of the following methods to verify success, depending on which migration method you used:

- If you used the runtime migration method, see Verifying migration.
- If you used the manual migration method, verify that your applications work as expected.
- If you used the artifact migration method, verify that your applications work as expected.

7. Rollback if necessary

If the migration was not successful, you may need to roll back your environment and perform the migration again. Use one of the following rollback methods, depending on which migration method you used:

- If you used the runtime migration method, see Rolling back your environment.
- If you used the manual migration method, you may need to uninstall and then reinstall the applications.
- If you used the artifact migration method, you may need to uninstall and then re-import and re-migrate the applications and source artifacts using WebSphere Integration Developer or WebSphere Business Modeler.

Migration methods

There are three types of version-to-version migration methods to choose from when considering moving up to a new version of WebSphere Process Server: runtime migration, manual migration, and artifact migration.

- “Runtime migration (production environment)”
- “Manual migration (parallel production environment)”
- “Artifact migration (parallel production environment with development tool migration)” on page 6

Runtime migration (production environment)

In production environments, the runtime migration procedures and tools can be utilized to migrate topology configuration, applications, and databases to the new version of WebSphere Process Server. The runtime migration procedures and tools support both stand-alone and network deployment environment migrations, as well as variants that include migration to a remote system (stand-alone environments only), migration while an operating system is being upgraded to a supported version (stand-alone environments only), and network deployment variants to support full downtime migration windows and minimal downtime migration windows. The runtime migration process replicates the source production configuration into the target environment. During the migration process, the target production environment replaces the source production environment, so the two environments are never operated in parallel.

The runtime migration procedures and tools should be used in the following scenarios:

- You want to move your applications to the new version without a dependency on the development tools and the development environment.
- You want to automatically have your source production environment configuration and applications replicated in the target production environment.
- You have long-running process and/or human task instances that have started in the source environment and need to complete in the target environment.
- You have product data in queues or failed events in product databases that were created in the source environment and need to survive the migration and be managed in the target production environment.
- You can tolerate a production environment downtime window to perform the migration.

The high level tasks involved in runtime migration are:

1. Install the new product version.
2. Backup all production profiles and databases.
3. Migrate each source environment profile to the target environment.
4. Migrate or upgrade the product databases.
5. Migrate the product database data.

For more information on the runtime migration procedures and tools, see the “Migration overview” on page 1 topic.

Manual migration (parallel production environment)

An alternative to using the migration procedures and tools is to use the manual version-to-version migration process. With the manual migration process, you are free to create a parallel target production environment that is configured from scratch differently than the source production environment. Applications can then be selectively redeployed from the source production environment to the target production environment. The redeployed applications create their own database tables and application data in the parallel production environment so they do not have access to the application data stored in the databases configured for the source production environment.

The manual runtime migration process should be used in following scenarios:

- You want to move your applications to the new version without depending on the development tools and the development environment.
- You want to reconfigure your topology as part of the process of migrating to the new version of WebSphere Process Server.
- You do not have long running process instances and human tasks, or you can run parallel production environments while you drain the process instances and human tasks in the source environment as new instances are started in the target production environment.
- You have application data in queues or failed events in product databases that were created in the source environment that can be managed to completion in the source production environment while new messages and events are routed in parallel to the target production environment.
- You cannot incur any downtime in your production environment and can concurrently manage parallel source and target production environments.
- You want to selectively redeploy applications from your source production environment to your target production environment.

The high level tasks involved in manual migration are:

1. Install the new product version.
2. Configure your desired parallel production environment.
3. Manually deploy applications from the source environment to the target production environment.
4. Optional: Run both environments in parallel so business process instances and human task instances that are in progress finish in the source environment and new instances start in the target environment.

Artifact migration (parallel production environment with development tool migration)

The artifact migration process is similar to the manual migration process in terms of the configuration of the parallel target production environment but instead of the applications being manually redeployed from the source environment directly into the target production environment they are imported into the development environment and migrated by the development tools. This results in applications whose artifacts are migrated to the new version enabling the applications to then be modified to exploit the new capabilities delivered by version 7.0. The application can then be tested and deployed to the parallel target production environment. Consistent with the manual migration process, when the applications are deployed to the target production environment, they create a new set of database tables, so they do not have access to the application data stored in the databases configured for the source production environment.

The artifact migration should be used in the following scenarios:

- You want to leverage the development tools and development environment to migrate the application artifacts to the new version and validate the compatibility of your applications.
- You want to leverage the development tools to update your applications to exploit new capability delivered by version 7.0.

- You want to reconfigure your topology as part of the process of migrating to the new version of WebSphere Process Server, or you can manually duplicate your source production environment configuration in your parallel production environment.
- You do not have long running process instances and human tasks, or you can run parallel production environments while you drain the process instances and human tasks in the source environment as new instances are started in the target production environment.
- You have application data in queues or failed events in product databases that were created in the source environment that can be managed to completion in the source production environment while new messages and events are routed in parallel to the target production environment.
- You cannot incur any downtime in your production environment and can concurrently manage parallel source and target production environments.
- You want to selectively migrate applications from your source production environment to version 7.0 with the development tools and selectively deploy those applications to your target production environment.

The high level tasks involved in artifact migration are:

1. Install the new product version.
2. Configure your desired parallel production environment.
3. Import the applications from the source production environment into development tools and migrate the applications according to the development tool's migration procedures.
4. Optional: Update the migrated applications to exploit new capability delivered in version 7.0.
5. Manually deploy the migrated applications from the development tools to the target production environment.
6. Optional: Run both environments in parallel so business process instances and human task instances that are in progress finish in the source environment and new instances start in the target environment.

For more information on artifact migration, see the migration section in the WebSphere Integration Developer and the WebSphere Business Modeler version 7.0 information center.

Migration method comparison

To determine the most appropriate migration method for migrating WebSphere Process Server to version 7.0, analyze the amount of stateful data in the environment, the amount of downtime your system can support, and whether you want to preserve your previous configuration.

Migration method considerations

There are several different issues to consider when determining what the right migration method is for migrating to version 7.0. The following section enumerates a set of items to consider when deciding what method best fits your migration requirements.

- Production data
- Downtime
- Long-running processes and human tasks
- Application enhancements

- Target environment configuration
- Risk mitigation
- Selective or phased application migration

Production data

The runtime migration method results in the source production environment being replaced by the target production environment. The implication on application data is that data that was created in the database by the source environment is available to the target environment post migration. This enables important scenarios. For example, processes and human tasks can be started in the source environment and finish in the target environment post migration. Messages in queues and failed events that existed in the source environment can be managed by the target environment post migration. The runtime migration method is the only method that provides this capability. The manual and artifact migration methods both result in a parallel production environment that has its own separate databases configured, completely distinct and independent of the source environment, even when the applications from the source environment are deployed to the target environment.

Downtime

The runtime migration method results in the source environment being replaced by the target environment while the manual and artifact migration processes depend on the creation of a parallel target environment. The implication is that the runtime migration method requires a downtime period when the databases are being upgraded and migrated from the source version to the target version prior to starting the migrated servers. The runtime migration procedures provide a minimal downtime procedure that can be used in some cases, but still does not eliminate the need for downtime.

The manual and artifact migration methods both require a parallel environment to be created that can be used in production concurrently with the source environment. The source and target environments can execute side-by-side until it is appropriate for the source environment to be discontinued. The ability to have two environments running concurrently on different versions also implies a level of operational complexity and likely requires additional capacity.

Long-running processes and human tasks

There are a few different scenarios and options regarding processes and human tasks to consider:

- The processes and tasks are short running and can be completed in the source environment just prior to the start of the migration downtime window

  If the migration process can incur downtime and the processes and tasks can be completed before the downtime window, all three of the migration methods are viable options. The decision of which option to use will thus depend on one of the other migration requirements.
- The processes and tasks are long-running and the migration can incur downtime

  In this scenario all three options are viable but there are important trade-offs to consider. Using the manual and artifact migration methods, the parallel production environments will need to be run concurrently for as long as it takes for the processes that started in the source environment to complete there. If a downtime window is not a gating factor, the runtime migration option is more ideal in this scenario enabling processes and tasks that are started in the source environment to complete in the target environment post migration.

- The migration cannot incur any downtime

  No downtime rules out the runtime migration method so either the manual or artifact migration method must be used to create a parallel target environment where the applications can be redeployed. Since these methods result in parallel environments that contain two different process and task databases, the new processes and tasks should ideally be started in the target environment, and the two environments must run in parallel until the processes and instances in the source environment have completed.

Application enhancements

The advantage of using artifact migration and the development tools is that the applications can be updated to the version 7.0 artifact level and the applications can then be enhanced with features provided in version 7.0.

Target environment configuration

If you require the same configuration in your target environment as your source environment then the runtime migration method is typically more appropriate since it will automatically replicate the source environment's topological configuration to the target environment. However, if you need to reconfigure the target environment configuration completely differently than your source environment for one of several good reasons, you must either do that before or after version-to-version migration as an independent exercise, or use either the manual or artifact migration methods if you plan to do it concurrent with the version-to-version migration.

Risk mitigation

The parallel environments provided by the manual and artifact migration methods enables a target production environment that is completely independent of the source environment that is serving the existing consumers enabling the target environment to be rigorously tested before going live in a production setting. In addition, artifact migration can reduce risk by leveraging the development tools to aid in verification that the application being migrated does not contain any issues that would present backwards compatibility challenges. Even in scenarios where migrations are leveraging the runtime or manual migration methods, artifact migration validation using the development tools is often done as an initial stage of the migration effort to validate application compatibility.

Selective or phased application migration

If you have a situation where you do not want to migrate all your applications in a single downtime window to the target version, you should use either the manual or artifact migration approaches. These approaches provide support for two parallel environments, the source and the target, and support selective or phased deployment of the migrated applications to the target environment. In contrast, the runtime migration method migrates all applications from the source environment to the target environment.

Migration method comparison

Use the following table to compare the benefits, costs, and risks of the three migration methods:

Table 1. Version-to-version migration methods: a comparison

Migration method	Benefits	Costs	Risks
Runtime migration	• No dependency on the development tools • Source environment configuration is replicated in the target environment • Source environment applications are migrated to the target environment • Source environment application data is moved, using existing database tables • Process and human tasks can start in the source environment and complete in the target environment • Application instance data on queues and failed events in the source environment can be handled post migration by the target environment • Additional hardware and/or software resources not required to manage another production environment	• Downtime is required when the target product environment assumes the role of the source production environment • Requires all applications on a node to be ready to migrate at the same time • New features are not enabled automatically and sometimes unavailable without migrating the application artifacts using artifact migration • Parallel production environment cannot be set up • Test focus: – End-to-end testing to validate migration process – Regression testing and performance tuning	• A rollback plan must be in place to handle a possible migration failure. For more information, see Rolling back your environment. • Existing user applications should continue to execute in the new runtime at the same level of function they had in the old runtime. In some cases, however, there may be a change in code on which the application depends, such as a JDK change, which may have negative impact on the unchanged application.

Table 1. Version-to-version migration methods: a comparison (continued)

Migration method	**Benefits**	**Costs**	**Risks**
Manual migration	• No dependency on the development tools • Target production environment can be configured differently than the source production environment since configuration is not automatically migrated from the source to the target • Parallel production environment supported: – Selective application migration – No downtime • Ability to perform extensive testing before migrating to production environment, but usually regression testing is enough • No dependency on migration tools	• Existing data is not moved; new database tables are created • New features are not enabled automatically and sometimes unavailable without migrating the application artifacts using artifact migration • Manual (scripted) deployment of applications is required • Requires updates to client applications • Hardware and software licenses may need to be evaluated for any additional licenses required when running in parallel	• Existing user applications should continue to execute in the new runtime at the same level of function they had in the old runtime. In some cases, however, there may be a change in code on which the application depends, such as a JDK change, which may have negative impact on the unchanged application.
Artifact migration	• Ability to exploit new features • Parallel production environment supported: – Selective application migration – No downtime • Ability to perform extensive testing before migrating to production environment • No dependency on migration tools	• New development environment is required • Existing data is not moved; new database tables are used • Manual (scripted) deployment of applications is required • Requires updates to client applications • Hardware and software licenses may need to be evaluated for any additional licenses required when running in parallel • Additional test coverage for application updates is required	• Application updates might require some level of testing.

Supported source migration paths

The following product and version combinations are supported as sources for version-to-version migrations to WebSphere Process Server version 7.0.

- WebSphere Process Server version 6.2.0.x
- WebSphere Process Server version 6.1.2.x
- WebSphere Process Server version 6.1.0.x
- WebSphere Process Server version 6.0.2.x

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 one of the versions that are supported migration sources using the manual migration method and then you can use the runtime migration method to migrate from that version to version 7.0.

Migration types

Runtime migration supports the migration of stand-alone environments and network deployment environments.

Stand-alone migration

Three types of stand-alone migration variants are supported by the runtime migration procedures and tools:

- **Side-by-side migration**: where the source and target of the migration are on the same system
- **Remote migration**: where the source and target of the migration are on are different systems
- **Operating system upgrade migration**: where the operating system on the source system is being upgraded during the migration procedure to a new version that is supported by WebSphere Process Server version 7.0.

The following sections describes each of these types of stand-alone environment migration variants in more detail.

Stand-alone side-by-side migration

The stand-alone side-by-side migration process is the simplest runtime migration scenario where the target product is installed on the same system as the source product, and the runtime migration procedures and tools are used to migrate the stand-alone profile containing the configuration, applications and the product databases to the target environment.

Stand-alone remote migration

The stand-alone remote migration process enables WebSphere Process Server version 7.0 to be installed on a different system than the source of the migration in order to support migration of the configuration and applications from one system to another. The stand-alone remote migration process can be used to support a variety of scenarios including:

- Migrating to a remote system that has the same type of hardware, operating system, and operating system version as the source of the migration
- Migrating to a remote system that has a different type of hardware (64-bit for example), a different operating system, or a different operating system version

The process requires the migration commands on the target system to be copied to the source system where they are used to copy the source profiles. The snapshot directory is then copied to the target system and used as the source for the profile migration.

Stand-alone operating system upgrade migration

The stand-alone operating system upgrade migration process enables the operating system on the system containing the source of the migration to be upgraded during the migration process. This is typically necessary if the operating system version containing the source product version is no longer supported by WebSphere Process Server version 7.0.

The process requires that you copy each of the source profiles on the prior version of the operating system, back up the copied source profiles to a remote location, reinstall the operating system to the new version, install the target product, restore the copied source profiles back to the migration system with the updated operating system, and then use the snapshot directory as the source for the profile migration.

Network deployment migration

Network deployment environment migrations are more involved than stand-alone environment migrations due to the need to migrate the deployment manager, clusters, nodes, and differently scoped product databases, in the appropriate order. All network deployment migrations require WebSphere Process Server version 7.0 to be installed side-by-side with the source product of the migration. If the source of the migration is augmented by additional BPM products, they should be installed in the same installation directory as WebSphere Process Server version 7.0.

Note: The remote migration and operating system upgrade migration variants supported in the stand-alone profile scenarios are not supported for network deployment migration.

Two different types of nodes are referred to in the network deployment migration procedures, **clustered nodes**, and **non-clustered managed nodes**. Clustered nodes contain at least one server that is a member of a cluster. Non-clustered managed nodes do not contain any servers that are cluster members.

Runtime migration tools

Migrating stand-alone and network deployment environments requires that you manage the production environment (start and stop the deployment manager, servers, and nodes), migrate configuration profiles, upgrade product databases, and migrate application data. The runtime migration procedures guide you through the process and the runtime migration tools are used to perform the required steps.

The following three sets of tools support the runtime migration procedures:

- “Profile migration tools” on page 14
- “Database upgrade and migration tools” on page 14
- “WebSphere Application Server management tools” on page 15

The following sections provide a summary of each of these groups of tools.

Profile migration tools

The profile migration tools are used to migrate the profiles that contribute to the cell, cluster(s), non-clustered managed node(s), or stand-alone server(s) being migrated.

The profile migration tools support a three step process for each profile:

1. Snapshot the configuration files from the source profile to be migrated
2. Create the target profile in the target installation using the snapshot configuration from the source profile
3. Migrate of the configuration snapshot to the target profile

The three step process required to migrate each profile is supported by the following set of profile migration command-line tools:

- BPMSnapshotSourceProfile command-line utility
- BPMCreateTargetProfile command-line utility
- BPMMigrateProfile command-line utility

In addition to the three step process for profile migration, the following command-line utilities play key roles in profile migration:

- The BPMCreateRemoteMigrationUtilities command-line utility creates an archive that can be copied to source migration systems to support remote migration of stand-alone profiles.
- The BPMMigrateCluster command-line utility is required in addition to the profile migration tools to migrate cluster profile configuration information in a network deployment environment.
- The BPMMigrateProfile command-line utility enables the optional migration of the Business Rules Manager in a network deployment environment. For more information about the Business Rules Manager, see installBRManager command-line utility.

For a summary of the profile migration commands see the “Runtime migration tools reference” on page 63 topic

Database upgrade and migration tools

WebSphere Process Server version 7.0 utilizes the following product databases that are either automatically or manually upgraded or migrated during the migration of the environment:

- Business Process Choreographer database
- Business Space database
- Common database
- Common Event Infrastructure database
- Messaging Engine database

The Common Event Infrastructure database and the Messaging Engine databases are both automatically migrated as needed by the profile migration process. The other database are either automatically or manually upgraded or migrated according to the detailed steps found in the migration procedures. When manually updating the product databases the commands and scripts for each of the supported database types must be invoked on the database system by a user with sufficient privileges or a system that has the database client utilities installed with a network connection to the database system. The runtime migration procedures describe how to copy the commands and scripts for your database type and the source release of the migration to the database system.

For a summary of the database migration commands see the “Runtime migration tools reference” on page 63 topic.

WebSphere Application Server management tools

During the migration procedures, the deployment manager, nodes, and servers must be stopped and started at various steps. In addition, there are several other WebSphere Application Server commands that are used throughout the migration procedures.

For a summary of the WebSphere Application Server management tools required by the migration procedures see the “Runtime migration tools reference” on page 63 topic.

Profiles

The runtime migration tools support the migration of WebSphere Process Server, WebSphere Enterprise Service Bus and WebSphere Application Server source profiles to the same profile type on the migration target.

WebSphere Process Server profile

A WebSphere Process Server profile is one that WebSphere Application Server created using one of the following profile templates: “default.wbiserver,” “dmgr.wbiserve,” or “managed.wbiserver.” When using the Profile Management Tool (PMT), this means that you select **WebSphere Process Server** on the Environment selection page

WebSphere Enterprise Service Bus profile

A WebSphere Enterprise Service Bus profile is one that WebSphere Application Server created using one of the following profile templates: “default.esbserver”, “dmgr.esbserver”, or “managed.esbserver.” When using the Profile Management Tool (PMT), this means that you select **WebSphere Enterprise Service Bus** on the Environment selection page.

WebSphere Application Server profile

A WebSphere Application Server profile is one that WebSphere Application Server created using one of the following profile templates: “default”, “dmgr”, or “managed.” When using the Profile Management Tool (PMT), this means that you select **WebSphere Application Server** on the Environment selection page.

Important: Even though the above definitions refer to the Profile Management Tool as a tool that may have been used to create the source profiles being migrated, you cannot use the Profile Management Tool or the manageprofiles command-line utility to create profiles that are the target of a migration with the exception of WebSphere Enterprise Service Bus profiles that are being migrated from 6.0.2.

The runtime migration procedures require the use of the BPMCreateTargetProfile command-line utility to create the migration target profiles. For example, if a system contains a WebSphere Enterprise Service Bus product installation that contains a WebSphere Enterprise Service Bus stand-alone profile created from the default.esbserver template, and the same system contains a WebSphere Process Server product installation that contains a WebSphere Application Server stand-alone profile created from the default template, a WebSphere Enterprise Service Bus stand-alone profile created from the default.esbserver template, and a WebSphere Process Server stand-alone profile created from the default.wbiserver template, all four stand-alone profiles are valid sources for migration to a WebSphere Process Server installation.

Deployment manager profile

In a WebSphere Process Server network deployment environment, the deployment manager must be created using the WebSphere Process Server deployment manager profile.

Product profile augmentation

The runtime migration tools support the migration of source profiles that have been augmented by one or more of the following BPM products:

- WebSphere Dynamic Process Edition
- WebSphere Business Services Fabric
- WebSphere Process Server
- WebSphere Enterprise Service Bus
- WebSphere Business Monitor
- WebSphere Business Compass

  Note: WebSphere Business Modeler Publishing Server changed to WebSphere Business Compass in version 7.0.

Augmented source profiles are migrated to a target profile that is augmented with the same product profiles so the target installation must have at least the same profile capabilities as the source.

For example, if a source installation contains a managed profile that has been augmented by WebSphere Process Server and WebSphere Business Monitor, the target installation directory must contain both WebSphere Process Server and WebSphere Business Monitor. In this scenario, the BPMCreateTargetProfile command-line utility will create a target profile that is augmented by WebSphere Process Server and WebSphere Business Monitor.

In a multi-product augmentation environment, where a cell may have clusters and nodes within profiles at various augmentation levels, the deployment manager profile must be augmented at the same augmentation level as the highest augmentation level of any of the profiles in the clusters or nodes.

Mixed-version environments

Version-to-version migration of network deployment-based production environments frequently results in a period of time when the network deployment environment is running applications on different versions of WebSphere Process Server. This concept is referred to as **mixed versions**.

Mixed versions of a product can theoretically be applied to multiple cells, mixed-version cells (multiple clusters or managed non-clustered nodes in a single cell), or mixed-version clusters (managed nodes in a single cluster). Only two of these types mixed versions are supported by WebSphere Process Server: **multiple cells** and **mixed-version cells**.

Multiple cells

If you have two cells that are initially at version 6.2.0, one can be upgraded to version 7.0 without having any administrative or database impact on the other cell. This is the simplest way to manage applications that are frequently running on different versions of WebSphere Process Server.

Mixed-version cells

In addition to having cells at different versions, clusters and non-clustered managed nodes in a single cell can be at different versions. For example, a cell might have one cluster at version 6.2.0 and another cluster that was at version 6.2.0 that has been migrated to version 7.0. In a mixed-version cell environment, the cell scoped Common database is being shared by all the clusters and non-clustered managed nodes that are running different versions of WebSphere Process Server.

Note: Mixed-version cell scenarios are not supported between version 6.2.0 and version 7.0 when both are supporting applications that use the Business Calendars capability.

If, during the course of migrating WebSphere Process Server to version 7.0, your cell is running nodes concurrently at the new and premigration levels, be aware that when the 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 levels:

- Configure Business Process Choreographer
- Install, update, or uninstall any applications that contain business processes, human tasks, or both

Mixed-version clusters

WebSphere Process Server does not support nodes in a single cluster running on different versions of WebSphere Process Server. This concept is referred to as a mixed-version cluster. 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 7.0 cluster member. Also, once the version 7.0 cluster member is started, the members of the cluster configured at a pre-version 7.0 level must not be started.

If, during the course of migrating WebSphere Process Server to version 7.0, your cell is running nodes concurrently at the new and premigration levels, be aware that 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.

Databases

WebSphere Process Server leverages several product databases during production that are either automatically migrated or must be manually migrated as part of the runtime migration procedure.

Database scopes

Some of the WebSphere Process Server product databases are cell scoped and others are cluster scoped.

The Common database is cell scoped, so any time any cluster or non-clustered managed node in the cell is migrated to version 7.0, the Common database must be migrated. In a mixed version cell environment, this may result in pre-version 7.0 clusters and non-clustered managed nodes utilizing the same instance of the Common database as version 7.0 clusters and non-clustered managed nodes.

The Business Process Choreographer database, Business Space database, Common Event Infrastructure database, and the Messaging Engine database are all cluster scoped. In a mixed version cell environment, each cluster or non-clustered managed node will have a unique instance of these databases if they are configured, and each instance will have schema and data that are unique to that version of the product. When each cluster or non-clustered managed node is migrated, its cluster scoped database is also migrated as part of the runtime migration procedures.

Backups

The migration procedures include steps for backing up the product databases to enable them to be restored if schema migration or data migration fails.

Automatic and manual migration

The Common Event Infrastructure database and Messaging Engine database are automatically migrated by the runtime migration procedure when the profiles are migrated. The Common database is automatically migrated in some situations as part of the runtime migration procedure and in other conditions manual migration is necessary. The Business Process Choreographer and Business Space databases requires manual migration in all circumstances. In summary, 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 a user ID with sufficient permissions for the Common database and the Business Process Choreographer database)
- If you used non-default table spaces
- If your migration source is configured with Business Space

More details on when and under what conditions the product databases should be manually migrated are captured directly in the runtime migration procedures.

Authorization

Because each of the database scripts require 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.

- **For the Business Process Choreographer database scripts:**

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

  CREATE BUFFERPOOL

  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.

If you are upgrading from version 6.0.2, 6.1.0, or 6.1.2, 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 i5/1 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.

- **For the Common database scripts:**

  The following permissions are required:

  CREATE TABLE

  ALTER TABLE

  DROP INDEX

  CREATE INDEX

  CREATE VIEW

  DROP VIEW

  CREATE SEQUENCE

- **For the Business Space database scripts:**

  The following permissions are required for all database types:

  ALTER TABLE

  CREATE TABLE

  INSERT

  CREATE INDEX

  Specific permissions for specific databases besides the ones mentioned for all databases are as follows:

  For DB2 for Linux, Unix and Windows

  CREATE BUFFERPOOL

  CREATE TABLESPACE

  For DB2iSeries:

  CREATE COLLECTION

  For DB2zOSV8 and DB2zOSV9:

  CREATE TABLESPACE

  For Oracle:

  CREATE TABLESPACE

  ALTER SESSION

  CREATE USER

  ALTER USER

GRANT

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.

DB2® for z/OS® and OS/390® Version 7

If you use 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, you will be asked to do that as part of the runtime migration procedure.

Oracle 9i and the Oracle JDBC driver

If you are using Oracle 9i, and have not yet upgraded your database to 10g or 11g, you will be asked to do that as part of the runtime migration procedure.

If you are using the Oracle ojdbc14.jar or the ojdbc5.jar JDBC driver, you will be asked to install and configure the ojdbc6.jar JDBC driver as part of the runtime migration procedure.

After data migration: Retuning your database and recreating custom views

During data migration, any additional indexes and custom views that you had are lost, and must be recreated.

Creating custom indexes is especially important for the performance of human workflow applications that make complex database queries.

Cloudscape to Derby Migration

In version 6.1.0 of WebSphere Process Server, the Cloudscape database was replaced with Derby. Under most circumstances, the profile migration tools automatically migrate Cloudscape databases to Derby. Exceptions are handled by the Migrating IBM Cloudscape or Apache Derby databases topic.

Downtime requirements

Stand-alone and network deployment migrations both require a period of time during which the applications are unavailable.

Runtime migration

All runtime migration method procedures require a period of downtime.

If downtime is not an option for the migration, consider the manual or artifact migration methods. See the Migration methods topic for more information.

Stand-alone environments

All three variants of the stand-alone migration procedure result in the stand-alone server being unavailable for the duration of the execution of the procedure.

Network deployment environments

Network deployment migration can be done by following a full downtime procedure or a minimal downtime procedure.

The network full downtime procedure assumes a migration downtime window where the network deployment environment is quiesced, all of the profiles are migrated, the database is upgraded, and the migrated version of the environment is started. The minimal downtime procedure allows half the nodes in a cluster to be migrated while the other half are servicing consumer requests, minimizing downtime to the period where the nodes running the prior version are shut down, the database is upgraded, and the migrated nodes are started. Use the full downtime procedure if the migration can be completed in the downtime window scheduled for the migration otherwise use the minimal downtime procedure.

What gets migrated

When you use the BPM runtime migration procedures to migrate to WebSphere Process Server version 7.0, the following items are migrated: user applications, adapters, profile configuration data, data sources and providers, and long-running processes.

User Applications

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

- **System Applications**: 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**: Sample applications are handled differently. For stand-alone profiles, the migration process does not install any sample applications. For network deployment profiles, any samples installed with the previous version of WebSphere Process Server will be installed during migration to the new version.

Business Rules Manager

The Business Rules Manager at any given version can manage applications containing Business Rules at the same version or a later version (in most cases) but does not support managing applications containing Business Rules created and deployed on prior versions. Since the Business Rules Manager is cell scoped, meaning it manages all Business Rules deployed in a cell, and cells can be mixed version, containing for example clusters on version 6.2.0 and clusters on 7.0, it is typically wise to defer the migration of the Business Rules Manager until all the Business Rules applications have been migrated. To support this concept, the Business Rules Manager application is not automatically migrated until the last non-clustered managed node or the last cluster in a cell is migrated.

Note: If the last migrated node is not a WebSphere Process Server profile, business rules resources and the Business Rules Manager migration script are not available. Therefore, Business Rules Manager is not automatically migrated during the migration process. In this scenario, you must manually run the Business Rules Manager migration script in a WebSphere Process Server custom node after the entire system is migrated. For more information, see installBRManager command-line utility.

As an example, assume a scenario where a cell contains four clusters named cluster1, cluster2, cluster3, and cluster4, each running version 6.2.0, the Business Rules Manager is deployed to cluster1, and you want to migrate the cell sequentially beginning with cluster1 followed by clusters 2, 3, and 4. If cluster1 is migrated first to version 7.0, the Business Rules Manager deployed to cluster1 remains at version 6.2.0 enabling it to continue to manage the Business Rule applications deployed to clusters 2, 3, and 4. The Business Rules Manager continues to run at the 6.2.0 version while clusters 2 and 3 are migrated, but it is then automatically migrated to version 7.0 when cluster4 is migrated.

There are also scenarios where it makes sense to manually migrate the Business Rules Manager at an earlier step instead of waiting for the very last node in the cell to be migrated. Take for example a slightly modified version of the prior scenario where the Business Rules Manager is deployed to cluster1, and only cluster2 contains Business Rule applications. Similar to the fist scenario, when cluster1 is migrated to version 7.0, the Business Rules Manager remains at the version 6.2.0, enabling it to manage the Business Rules deployed to cluster2. When cluster2 is migrated to version 7.0, it then makes sense to migrate the Business Rules Manager since clusters 3 and 4 do not contain any Business Rules, and the only rules in the cell are now at version 7.0. To support this scenario, the Business Rules migration process provides the installBRManager command-line utility that can be manually invoked at various phases of the migration process. For more information, see installBRManager command-line utility.

Note: In a stand-alone migration scenario, the Business Rules Manager is always automatically migrated when the stand-alone profile is migrated.

Adapters

For version 6.1.0, 6.1.2 and 6.2.0 WebSphere Adapters, you need to install an interim fix with the name "Mandatory adapter fix for running 6.1 and 6.2 Adapters on WPS v7.0". Please apply this interim fix on the source environment if you do not plan to update the WebSphere Adapter to a 7.0 level and want to use the application with 6.1.0 or 6.1.2 version of the Websphere Adapter.

Note: All WebSphere Adapters for version 6.0.2 and Websphere Adapter for SAP versions 6.0.2, 6.1.0, 6.1.2 and 6.2.0 are not supported on WebSphere Process Server version 7.0. This set of adapters has to be updated to version 7.0 before any applications using them can be deployed on WebSphere Process Server version 7.0. For more information specific to WebSphere Adapters, refer to the Websphere Integration Developer version 7.0 information center, or to “Runtime premigration checklist” on page 24.

Profile configuration data

The version-to-version migration tools (commands) automatically apply the configuration settings from the previous profile to the new profile created during the migration process.

JDBC providers and data sources

Profile migration automatically migrates the JDBC provider and data source definitions for each existing data source and provider.

Long-running processes

Long-running business process instances and human task instances are handled 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.

Note: Because the previously installed predefined human task applications might still have instances running, they are not uninstalled during migration. This means that after migration, both the new and the previous versions of the predefined human task applications are installed on your system. The version numbering indicates when the application was last updated. For information on when you can safely uninstall the previous versions of the applications, see “Postmigration tasks for Business Process Choreographer” on page 55.

What does not get migrated?

Certain artifacts do not get automatically migrated. Most of these artifacts are user-created, and are not recognized by WebSphere Process Server. Since they are not recognized, they are not migrated.

- **Share-by-reference (shared library) artifacts**

  If you are using the share-by-reference pattern for sharing SCA Libraries, then any artifacts that exist in the `lib/ext` and `config` directory, such as Java .jar libraries, are not migrated to the migration target. Although, the WebSphere configuration settings for share-by-reference libraries are transferred during profile migration, the actual library .jar artifact should be copied manually post-migration.
- **Most custom files or artifacts added to the WebSphere Process Server installation directory or profile directory structure**

  Most non-product files, such as custom Jython scripts, are not transferred as part of migration.

  Note: The only exception is that custom XSL transformation files for Business Process Choreographer are migrated automatically. These files are located in the *install_root*/ProcessChoreographer/Staff directory. For more information about these files, see “Postmigration tasks for Business Process Choreographer” on page 55.

  Similarly, if you have modified any WebSphere-specific scripts, then these changes need to be manually reapplied to the migration target after migration.

  Important: Keep any custom scripts or modified product scripts outside of the installation directory to prevent any accidental deletion of user-modified scripts.

Known compatibility issues

The following items are known compatibility issues when migrating to WebSphere Process Server version 7.0.

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

Note: This applies only if you have Business Process Choreographer configured. 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.

SCA Wiring

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 6.12 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 set the WebSphere variable SCA_USE_WS_FOR_DYNAMIC_INVOCATION to include the name of the modules in a semi-colon delimited list, for example, sca/myModule1;sca/myModule2

Runtime premigration checklist

Before you begin the process of migrating to a new version of WebSphere Process Server, you should verify each of the items in this checklist.

- "Hardware, operating system, and database prerequisites" on page 25
- "WebSphere Process Server install images" on page 25
- "DB2 for z/OS version 8 or 9 install images" on page 25
- "Upgrading Oracle database and JDBC driver" on page 25
- "Data direct driver bundled with WebSphere Application Server" on page 25
- "Uninstall applications using WebSphere Adapters, version 6.0.2.x" on page 26
- "Apply the WebSphere Adapter interim fix" on page 26
- "Uninstall Websphere Adapter for SAP applications and adapters" on page 27
- "Source profile backup directory storage" on page 27
- "Source database backup storage" on page 27
- "Source profile snapshot directory storage" on page 27
- "Target profile directory storage" on page 28
- "Business Process Choreographer data migration: Materialized views" on page 28
- "Ulimit setting" on page 29
- "Database authorization" on page 29
- "Determine the Appropriate Procedure and Procedure Variances" on page 29
- "Migrating root configurations to non-root" on page 29
- "Migrating non-root configurations to root" on page 29

Hardware, operating system, and database prerequisites

Verify that your target migration environment is a supported operating environment for WebSphere Process Server version 7.0. This includes the hardware platform, the operating system, and the database. For information on the supported operating environment for WebSphere Process Server version 7.0, see Preparing to install WebSphere Process Server.

WebSphere Process Server install images

Download the WebSphere Process Server install images and the latest fix packs so they are ready to be installed on each system to be migrated. Validate that there is sufficient storage on the system to have WebSphere Process Server and fix packs installed.

DB2 for z/OS version 8 or 9 install images

If you are using DB2 version 7 on your database server, download the install images for DB2 for z/OS version 8 or version 9 and be prepared to install them as a step in the procedures.

Upgrading Oracle database and JDBC driver

If you are using Oracle 9i, and have not yet upgraded your database to 10g or 11g, download the Oracle 10g or 11g install images and be prepared to upgrade to the new database version as a step in the procedures.

If you are using the Oracle ojdbc14.jar or the ojdbc5.jar JDBC driver, download the new ojdbc6.jar JDBC driver and be prepared to install and configure it as a step in the procedures.

Data direct driver bundled with WebSphere Application Server

The embedded data direct driver bundled with WebSphere Application Server is not supported with WebSphere Process Server version 7.0. You must either buy a license for the existing embedded data direct driver or download the Microsoft JDBC driver for MSSQL Server, which can be downloaded from Microsoft's website.

If the source version is 6.1.2 or 6.2.0 or you plan to buy the Data Direct driver, please update the existing data sources using the embedded data direct driver to use the new JDBC Driver in the source environment by performing the following steps.

1. Create a new Data Source with the correct JDBC Provider Type, and set the following properties: JNDI Name, statementCacheSize, releationalResourceAdapter, authMechanismPreference, authDataAlias, databaseName, serverName, portNumber, and URL that match with the existing Data Source.
2. Delete the existing Data Source that uses the embedded driver.
3. Test the connection for the Data Source.
4. Test the source environment that all applications continue to work.

If you are using source version 6.0.2 or 6.1.0 and you plan to use the Microsoft JDBC driver, copy the downloaded driver JAR files to the location where the previous driver JAR files are.

You will perform the Data Source configuration updates during migration procedure.

Uninstall applications using WebSphere Adapters, version 6.0.2.x

All Websphere Adapters version 6.0.2.x are not supported by WebSphere Process Server version 7.0 due to WebSphere Adapters support on runtime for N-2 releases. For more information on updating the applications and using the 7.0.x versions of Websphere Adapters, see Migrating applications using previous adapter levels on the WebSphere Integration Developer information center.

Apply the WebSphere Adapter interim fix

If any of the applications in the source environment embed any of the WebSphere Adapters (with the exception of SAP) at either version 6.1.0 or version 6.2.0 or use WebSphere Adapter versions 6.1.0 or 6.2.0 configured at the node level, then an adapter interim fix must be applied in the source environment prior to starting the migration procedure. This can be done as follows:

1. Obtain the "Mandatory adapter fix for running 6.1 and 6.2 Adapters on WPS v7.0" for the WebSphere Adapter(s) your applications utilize. Use one of the options below to obtain the iFix:
   - If you are using Websphere Adapter version is 6.2.x or 6.1.x, contact IBM support team to obtain the corresponding Adapter IFix.
   - If you are using WebSphere Adapter version 6.1.x, update WebSphere Integration Developer and extract the RAR file. Use the following procedure.
     a. Download the following version of WebSphere Integration Developer: WebSphere Integration Developer version 6.1.2 Interim Fix 005.
     b. Update your existing WebSphere Integration Developer to the new version using Install Manager.
     c. Extract the RAR file from the following directory: `WebSphere Integration Developer/`*`installation_directory`*` /ResourceAdapters`.
2. Apply this fix over the respective WebSphere Adapter on the source version. Use one of the following procedures, depending on whether the WebSphere Adapter is embedded in the application or installed at the node level.
   - If the WebSphere Adapter is embedded in the application, use the following procedure.
     a. Log in to the administrative console.
     b. Select the Application, then click **Update**.
     c. Select the **Single Module** with a relative path as name of the WebSphere Adapter RAR file.
     d. Click **Browse** to select the updated RAR file on the local file system that has the changes.
     e. Select the default values in the remaining steps, then click **Finish**. This will ensure that existing configurations, for example, are not changed and that only JAR files will be updated.
   - If the WebSphere Adapter is installed at the node level, use the following procedure.
     a. From the administrative console , browse the WebSphere Adapter instance and make note of all Managed Connection Factory and ActivationSpec instances configured for the adapter.
     b. Select the WebSphere Adapter, then click **Delete** to uninstall the adapter.
     c. Install the new version of the WebSphere Adapter.

d. Configure the Managed Connection Factory and ActivationSpec instance that you made note of in Step a.

Note: If the dependent applications have configuration for ManagedConnectionFactory and ActivationSpec in the.import and .export files respectively, you can also uninstall and install the application to recreate the configuration for ManagedConnectionFactory and ActivationSpec. If the Application uses JNDI reference to configure the ManagedConnectionFactory and ActivationSpec, you must manually recreate the instances as documented in the steps above.

Uninstall Websphere Adapter for SAP applications and adapters

All pre-version 7.0 versions of the WebSphere Adapter for SAP are not supported by WebSphere Process Server version 7.0 due to incompatible changes introduced by the SAP SAPJCO library updated to support the Java Runtime Environment version 1.6. The lack of compatibility necessitates that all the applications that use the WebSphere Adapter for SAP are manually uninstalled from the source environment prior to executing the runtime migration procedure, updated in WebSphere Integration Developer, and redeployed manually to the target production environment post migration. In addition, all WebSphere Adapters for SAP installed at a node level need to be removed from the source environment prior to executing the runtime migration procedure and the new version of the adapter configured manually in the target environment after runtime migration is complete.

For more information on updating the applications to support the new SAP SAPJCO library and the WebSphere Adapter for SAP version 7.0, see WebSphere Adapter for SAP Software documentation on the WebSphere Integration Developer information center.

Source profile backup directory storage

During migration the profile being migrated is backed up in case a roll back is necessary at some later point. The space available for the profile backup directory should be at least the size of the source profile's configuration directory and applications.

Source database backup storage

The migration procedures strongly recommend backing up your source product databases prior to migrating them. Verify that sufficient space exists to backup these databases. The size required for the backups will depend on the size of your production databases and the specifics of your database backup strategy.

Source profile snapshot directory storage

The configuration files in the profile to be migrated are copied during the migration procedure to a snapshot directory that then becomes the source for the profile migration. The directory is an optional parameter for the BPMSnapshotSourceProfile command and is defaulted to MigrationSnapshots.

Prior to migration, verify that sufficient storage for the snapshot directory exists. The storage requirements can be estimated by summing up the following amounts:

- Size of the profile configuration information to be migrated:

- *profile_root*/installableApps directory
- *profile_root*/installedApps directory
- *profile_root*/config directory
- *profile_root*/properties directory

- Size of the shared libraries to be migrated:
  - Shared libraries referenced in the `libraries.xml` configuration files
- Size of the resource adapter archives to be migrated:
  - Resource Adapter Archive (RAR) files referenced in the `resources.xml` configuration files
- If trace is enabled, allocate an additional 200 MB (depending on the size and complexity of your configuration) for the trace file written to the snapshot directory.

Target profile directory storage

During migration the target profile is created by using the BPMCreateTargetProfile command and the source profile is migrated to the target profile referenced from the target installation.

Prior to migration, verify that sufficient storage for the target profile directory exists. The storage requirements can be estimated by summing up the following amounts:

- Size of the profile configuration information to be migrated:
  - *profile_root*/installableApps directory
  - *profile_root*/installedApps directory
  - *profile_root*/config directory
  - *profile_root*/properties directory
- Size of the shared libraries to be migrated:
  - Shared libraries referenced in the `libraries.xml` configuration files
- Size of the resource adapter archives to be migrated:
  - Resource Adapter Archive (RAR) files referenced in the `resources.xml` configuration files
- If trace is enabled, allocate an additional 200 MB (depending on the size and complexity of your configuration) for the trace file written to the snapshot directory.

Business Process Choreographer 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 can re-create the named materialized views only if the **customTableDefinition** points to a custom table definition XML file that it can access. To verify that WebSphere Process Server can re-create your named materialized views perform the following:

1. Make sure that 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.

Ulimit setting

On UNIX systems, to avoid an error during profile migration due to too many open files, increase the ulimit setting on the system running the profile migration process.

Database authorization

Verify whether you will be able to run all the database scripts using a single user ID, or whether your database administrator might have to run any of them.

See the information in the Databases topic for more information on the required permissions for the product databases.

Determine the Appropriate Procedure and Procedure Variances

If you are migrating a stand-alone profile, determine whether you plan to do a side-by-side migration or a migration to a remote system. If you are migrating a network deployment environment, analyze the full downtime and minimal downtime procedures carefully to determine which procedure more closely fits your requirements.

Migrating root configurations to non-root

If you are migrating a previous version environment with root user permissions to version 7.0 with non-root user permissions, complete the steps in the Migrating root configurations to non-root topic on the WebSphere Application Server Version 7.0 information center before attempting the migration procedure.

Note: The reference to *USER_HOME* in the "Migrating root configurations to non-root" instructions refers to the *USER_INSTALL_ROOT* or the root directory of the source profile.

Migrating non-root configurations to root

If you are migrating a previous version environment with non-root user permissions to version 7.0 with root user permissions, complete the steps in the Migrating non-root configurations to root topic on the WebSphere Application Server Version 7.0 information center before attempting the migration procedure.

Runtime migration procedures

Use the runtime migration procedures to perform a version-to-version migration.

Migrating stand-alone servers

You can migrate a stand-alone environment by either using the supplied script, or manually. The migration process does not automatically back up the files on the source server, so it is good practice to back up the files on the source server before running the migration.

Before you begin

- Ensure that the server you are migrating from (source server) is a z/OS® stand-alone server with WebSphere Process Server version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 installed.
- Ensure that the server you are migrating to (target server):
  - is a z/OS stand-alone server with WebSphere Process Server version 7.0 installed and configured.
  - has been augmented to use the same database as the source server.
  - has been augmented with the response file keyword -createDefaultProfileForMigration set to `true`. Failure to do so will result in duplicate CEI messaging engines. See the Response file values topic for more information about this keyword.

Procedure

1. Stop the version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 server. See Stopping stand-alone servers.
2. Back up the WebSphere Process Server database. If necessary, you can then recover the version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 data later.
3. Generate the WebSphere Application Server migration jobs from the z/OS migration management tool component of the WebSphere Customization Tools.
4. In the installed WebSphere Process Server JCLPDS(ZWPS.*.*.SBPZJCL), locate the sample WebSphere Process Server migration jobs BPZWMG1B, BPZWMG2B, and BPZWMG3B and take a copy.

   The BPZWMG1B and BPZWMG2B jobs are only required if you had XA Connectors installed on the version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 server. The BPZWMG3B job is mandatory and performs the actual migration.
5. Edit the jobs so that they make use of the parameters generated in the previous step by the WebSphere Customization Tools.
6. Back up the files on the source WebSphere Process Server server.
7. Migrate the server. The migration process populates a temporary backup directory using the information it finds in the file system of the source server, and then uses this temporary backup directory to update the file system of the target server. Use the following procedure to migrate the server.
   a. If XA connectors were installed on the version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 server, run the BPZWMG1B and BPZWMG2B jobs.
   b. Run the BPZWMG3B job.

      Note: The BPZWMG3B job runs the wbimgrt2.sh script which starts the migration utilities BPMSnapshotSourceProfile.sh and BPMMigrateProfile.sh. As part of wbimgrt2.sh processing, the WebSphere Application Server profile is also migrated.

8. Verify the migration. The migration process produces numerous diagnostic log files that must be checked, including the following files:
   - All of the log files in the `/tmp/migration/nnnnnn` directory that you specified in the migration job.
   - In the event of failure, the log files in the migrated server's logs directory provide details. For example: `/WebSphere/V61T2Z1/AppServer/profiles/default/logs`.

   Important: Most of these files are generated as ASCII files, so you must convert them to EBCDIC if you want to view them from TSO. If the tools that you use to view, edit, and run the scripts require the scripts to be in EBCDIC format, use the iconv command to convert the file to EBCDIC. For example:

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

   The profile migration is now complete.
9. Upgrade the WebSphere Process Server database. You can do this manually, or by using the DbDesignGenerator.sh and upgradeDB.sh script scripts.
   a. To upgrade manually, use the following procedure:
      1) Copy the database migration upgrade scripts to your working directory. These scripts are generated in database specific directories within the `WebSphere servername/AppServer/dbscripts` directory of the target server.
      2) Check the database specific directories for any SQL files starting with the name *upgrade*, the version number of the source server, and either *wbi* for WebSphere Process Server or *esb* for WebSphere Enterprise Service Bus. The following example is a list of SQL files that were generated during a migration from a version 6.1.2 WebSphere Process Server stand-alone server that used DB2 v9:

         ```
         /WebSphere/V7T2Z1/AppServer/dbscripts/CommonDB/DB2zOSV9/upgradeSchema612.sql
         /WebSphere/V7T2Z1/AppServer/dbscripts/CommonDB/DB2zOSV9/upgradeSchema612_CommonDB.sql
         /WebSphere/V7T2Z1/AppServer/dbscripts/CommonDB/DB2zOSV9/upgradeSchema612_DirectDeploy.sql
         /WebSphere/V7T2Z1/AppServer/dbscripts/CommonDB/DB2zOSV9/upgradeSchema612_governancerepository.sql
         /WebSphere/V7T2Z1/AppServer/dbscripts/CommonDB/DB2zOSV9/upgradeSchema612_relationshipService.sql
         /WebSphere/V7T2Z1/AppServer/dbscripts/CommonDB/DB2zOSV9/wbiserver_upgradeSchema612_Recovery.sql
         /WebSphere/V7T2Z1/AppServer/dbscripts/BusinessSpace/DB2zOSV9/createDatabase.sql
         /WebSphere/V7T2Z1/AppServer/dbscripts/BusinessSpace/DB2zOSV9/createStorageGroup.sql
         /WebSphere/V7T2Z1/AppServer/dbscripts/BusinessSpace/DB2zOSV9/createTable.sql
         /WebSphere/V7T2Z1/AppServer/dbscripts/BusinessSpace/DB2zOSV9/createTables_BusinessSpace.sql
         /WebSphere/V7T2Z1/AppServer/dbscripts/BusinessSpace/DB2zOSV9/createTablespace.sql
         /WebSphere/V7T2Z1/AppServer/dbscripts/BusinessSpace/DB2zOSV9/dropTable.sql
         /WebSphere/V7T2Z1/AppServer/dbscripts/ProcessChoreographer/DB2zOSV9/upgradeSchema612.sql
         /WebSphere/V7T2Z1/AppServer/dbscripts/ProcessChoreographer/DB2zOSV9/upgradeTablespaces612.sql
         ```

      3) Copy the scripts to your working directory and assign the appropriate permissions.
      4) Edit the values in the file to suit your requirements. Remember to convert them from ASCII to EBCDIC as necessary.
      5) Run the customized scripts against the database using tool such as DBUtility.sh or SPUFI, or in a batch job. See Creating the DB2® databases and storage groups using SPUFI, DSNTEP2, or DButility.sh for more information.
   b. To upgrade the commonDB, BPC and BSpace databases using the DbDesignGenerator.sh and upgradeDB.sh scripts, use the following procedure:
      1) Run DbDesignGenerator.sh to create upgrade ddl for CommonDB (option 2 followed by option 5)
      2) Run DbDesignGenerator.sh to create upgrade ddl for BPC (option 2 followed by option 1)

3) Run DbDesignGenerator.sh to create upgrade ddl for BSpace (option 2 followed by option 3)

4) Run upgradeDB.sh script to run the upgrade ddl generated by DbDesignGenerator.sh.

10. Migrate Business Process Choreographer runtime data. After migrating a server that has Business Process Choreographer configured, you must perform a data migration before you start the server. This involves running the migrateDB.py script.
11. If you have used the same procnames for your source and target servers, update the started task JCL members in USER.PROCLIB by running the BBODCPY1 job from the jcl library that was generated during the install process of your WebSphere Application Server target server. This job replaces the version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 started task members with the new version 7.0 members.
12. Start the target server.

Results

The stand-alone environment is migrated to the target version.

What to do next

Verify that the migration was successful. For instructions, see "Verifying migration" on page 47.

Migrating a network deployment environment

You can migrate a network deployment environment by either using the supplied script, or manually. The migration process does not automatically back up the files on the source server, so it is good practice to back up the files on the source server before running the migration.

Before you begin

Before you migrate a network deployment environment you must install and configure a version 7.0 WebSphere Process Server network deployment configuration of the same type on z/OS. The newer version (version 7.0) configuration must have been augmented to use the same database as the older version (version 6.2.0, 6.1.2, 6.1.0, or 6.0.2) configuration uses. It is only neccessary to install and augment the deployment manager and managed nodes in the version 7 configuration. Do not federate any of the nodes or configure any other components, such as Business Process Choreographer.

Important: Use the same cell name and node name when migrating from version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 to version 7.0 . If you use different cell names and node names, federated nodes cannot successfully migrate to the version 7.0 cell.

You must also use the same procedure names for the daemon and controller regions.

The version 7 daemon and controller region procedure names do not have to be the same as those in the older configurations.

The target deployment manager nodes should not be federated.

About this task

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

This information is applicable if you are migrating a version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 node to a version 7.0 node.

You can migrate a WebSphere Process Server node that belongs to a cell to version 7.0 without removing the node from the cell.

Migrating a base WebSphere Process Server version 7.0 node that is within a cell also migrates the node agent to version 7.0. A cell can have some version 7.0 and other nodes that are at version 6.2.0, 6.1.2, 6.1.0, or 6.0.2.

Note: If, before migration, Business Rules Manager is running on any of the deployment targets (servers or clusters), you should review the following topic before starting to migrate: installBRManager command-line utility.

Migrating a deployment manager

You can migrate a deployment manager by either using the supplied script, or manually. The migration process does not automatically back up the files on the source server, so it is good practice to back up the files on the source server before running the migration.

Procedure

1. Stop the older version (version 6.2.0, 6.1.2, 6.1.0, or 6.0.2) deployment manager. See Stopping a deployment manager.
2. Back up the WebSphere Process Server database. If necessary, you can then recover the version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 data later.
3. Generate the WebSphere Application Server migration jobs and variables from the z/OS migration management tool component of the WebSphere Customization Tools.

   Note: Only the variables created by the WebSphere Customization Tools will get used, the jobs will not get run.

   To open this component, select the **Window** menu, **Open Perspective**, **z/OS Migration Management Tool** and complete the panels as appropriate to your configuration.
4. In the installed WebSphere Process Server JCL PDS(.**.SBPZJCL), locate the sample WebSphere Process Server migration jobs BPZWMG1D BPZWMG2D and BPZWMG3D and take a copy. The BPZWMG1D and BPZWMG2D are only required if you had XA Connectors installed in the version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 server. The BPZWMG3D job runs the wbimgrt2.sh script which starts the migration utilities BPMSnapshotSourceProfile.sh and BPMMigrateProfile.sh. As part of wbimgrt2.sh processing, the WebSphere Application Server profile also gets migrated.
5. Edit your copy of the sample jobs to make use of the parameters generated in step 3 by the WebSphere Application Server customization panels.
6. Submit the sample jobs. You must use the administrator user name and password when submitting the jobs.
7. Examine the output in `/tmp/migrate/xxxxx/BPZWMGxD.out` and `/tmp/migrate/xxxxx/dmgr_backup/logs` to ensure the migration completed successfully.

8. Upgrade the commonDB database. You can do this by using the DbDesignGenerator.sh and upgradeDB.sh script scripts or manually.
   a. To upgrade using the DbDesignGenerator.sh and upgradeDB.sh scripts, use the following procedure:
      1) Run DbDesignGenerator.sh to create the upgrade ddl for CommonDB (option 2 followed by option 5)
      2) Run upgradeDB.sh script to run the upgrade ddl generated by DbDesignGenerator.sh.
   b. To upgrade manually, use the following procedure:
      1) Copy the database migration upgrade scripts to your working directory. These scripts are generated in database-specific directories within the `/WebSphere/servername/DeploymentManager/dbscripts` directory of the target server.
      2) Check the database-specific directories for any SQL files containing the name *upgrade*, the version number of the source server, and either *wbi* for WebSphere Process Server or *esb* for WebSphere Enterprise Service Bus. The following example is a list of SQL files that were generated during a migration from a version 6.1.2 WebSphere Process Server stand-alone server that uses DB2 v9:

```
/WebSphere/V7T2Z1/DeploymentManager/dbscripts/CommonDB/DB2zOSV9/upgradeSchema612.sql
/WebSphere/V7T2Z1/DeploymentManager/dbscripts/CommonDB/DB2zOSV9/upgradeSchema612_CommonDB.sql
/WebSphere/V7T2Z1/DeploymentManager/dbscripts/CommonDB/DB2zOSV9/upgradeSchema612_DirectDeploy.sql
/WebSphere/V7T2Z1/DeploymentManager/dbscripts/CommonDB/DB2zOSV9/upgradeSchema612_governancerepository.sql
/WebSphere/V7T2Z1/DeploymentManager/dbscripts/CommonDB/DB2zOSV9/upgradeSchema612_relationshipService.sql
/WebSphere/V7T2Z1/DeploymentManager/dbscripts/CommonDB/DB2zOSV9/wbiserver_upgradeSchema612_Recovery.sql
```

      3) Copy the scripts to your working directory and assign the appropriate permissions.
      4) Edit the values in the file to suit your needs. Remember to convert them from ASCII to EBCDIC as necessary.
      5) Run the customized scripts against the database using the tool of your choice. For example, DBUtility.sh, SPUFI, or in a batch job. See Creating the DB2 databases and storage groups using SPUFI, DSNTEP2, or DButility.sh for more information.
9. If you have used the same procnames for your source and target servers, update the started task JCL members in USER.PROCLIB by running the BBODCPY1 job from the jcl library that was generated during the install process of your WebSphere Application Server target server. This job replaces the version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 started task members with the new version 7.0 members.
10. Start the deployment manager. See Starting deployment managers.

Results

The deployment manager is migrated to version 7.0.

What to do next

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

Migrating a managed node

You can migrate a managed node by either using the supplied script, or manually. The migration process does not automatically back up the files on the source server, so it is good practice to back up the files on the source server before running the migration.

Before you begin

Before migrating any managed nodes in the cell, you must first migrate the deployment manager. See Migrating a deployment manager.

Procedure

1. Stop the version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 managed node server but ensure that the deployment manager is running. See Stopping managed servers for more information.
2. Generate the WebSphere Application Server migration jobs and variables from the z/OS migration management tool component of the WebSphere Customization Tools.

   Note: Only the variables created by the WebSphere Customization Tools are used, the jobs are not run.
3. In the installed WebSphere Process Server JCL PDS(.**.SBPZJCL), locate the sample WebSphere Process Server migration jobs BPZWMG1F, BPZWMG2F, and BPZWMG3F and take a copy. The BPZWMG1F and BPZWMG2F jobs must be run only if you had XA Connectors installed in the version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 server. The BPZWMG3F job runs the wbimgrt2.sh script which starts the migration utilities BPMSnapshotSourceProfile.sh and BPMMigrateProfile.sh. As part of wbimgrt2.sh processing, the WebSphere Application Server profile is also migrated.
4. Edit and submit your copy of the sample jobs to make use of the parameters generated in step 2 by the WebSphere Application Server Customization panels. Make sure that you use the administrator user name and password when submitting the jobs.
5. Examine the output in `/tmp/migrate/xxxxx/BPZWMGxF.out` and `/tmp/migrate/xxxxx/fed_backup/logs` to ensure the migration completed successfully.
6. Migrate the cluster. Run the BPMMigrateCluster script from the target deployment manager bin directory, for example: `BPMMigrateCluster.sh /tmp/migrate/wpsdmgr612_3/dmgr_backup PN9Cluster1 default`. This step only has to be done once, not for every node, and only when half the nodes have migrated. This step has to be done before any migrated node can be started.
7. To update the BPC and BSpace databases using the DbDesignGenerator.sh and upgradeDB.sh script scripts, use the following procedure. You will only have to do this for the first node migrated in the cluster, not for every node:
   a. Run DbDesignGenerator.sh to create ddl for BPC (option 2 followed by option 1)
   b. Run DbDesignGenerator.sh to create ddl for BSpace (option 2 followed by option 3)
   c. Run upgradeDB.sh to run the upgrade ddl generated by DbDesignGenerator.sh.

   See Migrating a deployment manager for more information.
8. Migrate the Business Process Choreographer run-time data. After migrating or upgrading a server or cluster that has Business Process Choreographer configured, you must perform a one time data migration of the run-time data before you start the migrated server or any migrated cluster member. This involves running the migrateDB.py script from one of the cluster node's bin directory. For example:

```
wsadmin.sh -conntype NONE -profileName default -tracefile
migrateDB.py_trace -f /wps_install_dir/ProcessChoreographer/admin/
migrateDB.py -cluster ClusterT4 -dbSchema T4CELL -dbUser dbuser
-dbPassword dbpswd
```

9. If you have used the same procnames for your source and target servers, update the started task JCL members in USER.PROCLIB by running the BBOMCPY1 job from the jcl library that was generated during the install process of your WebSphere Application Server target server. This job replaces the version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 started task members with the new version 7.0 members.
10. Start the version 7.0 managed node server. See Starting a server from the MVS console for more information.

Results

The managed node is migrated to the target version.

What to do next

Verify that the migration was successful. For instructions, see "Verifying migration" on page 47.

Runtime migration subprocedures

Use the runtime migration subprocedures as part of the process of performing a version-to-version migration.

Migrating databases

After migrating a server or cluster, you must manually upgrade the schema for the Common database, Business Process Choreographer database, and Business Space database and possibly perform a data migration before you start the server or any cluster member. You can optionally use the upgradeDB.sh script to perform any combination of the manual database upgrades simultaneously.

The Common Event Infrastructure database and Messaging Engine database are automatically migrated by the runtime migration procedure when the profiles are migrated. For more information, see "Databases" on page 17.

Upgrading the databases using the upgradeDB.sh script:

You can use the upgradeDB.sh script to upgrade the WebSphere Process Server databases: the Common database, and the Business Process Choreographer database.

Before you begin

Before you run the upgradeDB.sh script, you must complete the following steps:

- Migrate the deployment manager or stand-alone server as described in Migrating a deployment manager and Migrating stand-alone servers.
- Run the DbDesignGenerator.sh tool to generate the Common database and BPC database migration DDL. The Common database migration DDL is generated by configuring the Common database; the BPC database migration DDL is generated by selecting a special migration option. The generated DDL should be stored in the default locations. For example, for a DB2 for z/OS V8 database migration the locations would be `/WAS_HOME/util/dbUtils/WBI_CommonDB_DB2-zOS-8-CommonDB` and `/WAS_HOME/util/dbUtils/DB2-zOS-8-BPC`.

About this task

Any database that is accessed by a migrated server must 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 must have its schema updated before you start any of the cluster members. You can use the upgradeDB.sh script to upgrade the databases. The userid that runs upgradeDB.sh must be authorized to perform all of the following operations:

- Create and alter tables
- Create and drop indexes and views.

Procedure

1. On the stand-alone server node or the deployment manager node, access the USS command shell and switch to the administrator user ID, or another userid that has write access to the configuration file system. For example:
   ```
   su wsadmin
   ```
2. Copy the sample upgradeDB.sh file to your working directory. For example:
   ```
   cp /usr/lpp/zWPS/V7R0/zos.config/samples/upgradeDB.sh /u/work
   ```
3. Assign the appropriate permissions to the copy of the upgradeDB.sh file:
   ```
   chmod 755 upgradeDB.sh
   ```
4. Customize the parameters in the copy of the upgradeDB.sh file as required by your system. See the upgradeDB.sh script topic for more information. Set the following WAS_HOME environment variables appropriately: For example:
   ```
   export WAS_HOME=/WebSphere/V6S01Z1/AppServer
   ```

   If necessary set your LIBPATH and STEPLIB variables in order to access DB2 code, For example:
   ```
   export LIBPATH=/ZOS180/usr/lpp/db2910/lib:$PATH
   export STEPLIB=SYS2.DB2.V910.SDSNEXIT:SYS2.DB2.V910.SDSNLOD2:SYS2.DB2.V910.SDSNLOAD:$STEPLIB
   ```
5. Save the edited file.
6. Run the customized upgradeDB.sh script using the following syntax:
   ```
   upgradeDB.sh -SourceVersion (version number)
   ```

   You can also specify additional parameters. See the topic upgradeDB.sh script for more information.
7. Check the error.out file for any errors in the tmp directory that is created by the upgradeDB.sh script in your `/u/work` directory.

Results

The WebSphere Process Server databases have been upgraded.

What to do next

Use the migrating topics in the **'Before you begin'** section to complete your migration.

Upgrading the Common database schema:

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

See the “Migrating stand-alone servers” on page 30 and “Migrating a deployment manager” on page 33 topics.

About this task

This procedure supports the Common database upgrade for the following database types:

Database type	Directory name
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` In version 6.1.0 of WebSphere Process Server, the Cloudscape database was replaced with Derby. Under most circumstances, the profile migration tools automatically migrate Cloudscape databases to Derby. Exceptions are handled by the Migrating IBM Cloudscape or Apache Derby databases topic.

Procedure

Use the following procedure to upgrade to a new database schema for the Common database.

1. On the database system, invoke the upgradeSchema command in either interactive mode where it prompts for its parameters or non-interactive mode where the parameters are specified on the command line.

   For more information about the Common database upgradeSchema command, see upgradeSchema command-line utility for the Common database.

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.

Upgrading the Business Process Choreographer database schema:

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

Before you begin

See the "Migrating stand-alone servers" on page 30 and "Migrating a deployment manager" on page 33 topics.

About this task

This procedure supports the upgrade of the Business Process Choreographer database:

Procedure

1. During migration, a database design file is generated. You must customize the database design file before you can use it to generate the script or scripts required to upgrade your schema.
   a. Locate the generated database design file. `*profile_root*/dbscripts/ProcessChoreographer/*database_type*/*database_name*/*database_schema*_createSchema.properties`. Where

      profile_root
      - If Business Process Choreographer is configured on a server, this is the profile of the corresponding node.
      - If Business Process Choreographer is configured on a cluster, this is the profile where you run BPMMigrateCluster (previously known as WBIProfileUpgrade.ant), which is normally the deployment manager profile.

      database_name
      is the name of the database.

      database_schema
      is the name of the database schema. It is optional, and is not set if an implicit schema is used.

      database_type
      is the name of the directory that corresponds to the database type that you are using.

   b. Make a copy of the appropriate database design (createSchema.properties) file.
   c. Start the database design tool to edit the database configuration that is defined in the copy of the properties file.
      1) Enter the following command:

      ```
      install_root/util/dbUtils/DbDesignGenerator.sh
                -e copy_of_createSchema.properties_file
      ```

      2) Answer all the questions, or press enter to accept the default values. In particular, make sure that you select the migration scenario and that the database name, the database schema qualifier, and any tablespace names are correct.
      3) You have the choice whether to overwrite the input file or to save your changes in a new file.
   d. Run the database design tool on your modified database design file to generate the upgrade scripts.

      Enter the following command:

```
install_root/util/dbUtils/DbDesignGenerator.sh
      -g copy_of_createSchema.properties_file
     [-d output_directory]
```

If you do not provide the `-d` option to specify an output directory, the generated files will be written to a subdirectory of the current directory.

- The tool generates `upgradeSchema`*`schema_version`*`.sql` for all schema versions from which you can migrate to this version.
- If your database uses table spaces, the tool will also generate some `upgradeTablespaces`*`schema_version`*`.sql` scripts.
- If you use DB2 and are migrating from a version earlier than 6.2, the tool will create `upgradeTablespace`*`schema_version`*`.sql` scripts that will create 8k table spaces.

For example, the following files might be generated:

```
upgradeSchema602.sql
upgradeSchema610.sql
upgradeSchema612.sql
upgradeSchema620.sql
upgradeTablespace602.sql
upgradeTablespace610.sql
upgradeTablespace612.sql
```

2. If you will run the script on a different system, copy the appropriate generated upgrade scripts to the system on which the database is hosted. You only need to copy the one or two scripts that match the *schema_version* that you are migrating from. For example, If you are migrating from version 6.2, copy the file `upgradeSchema620.sql`.
3. 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.
4. If there was a `createTablespace`*`schema_version`*`.sql` file generated for the version that you are migrating from, run it to create the 8k table spaces. For information about how to run an SQL script on your database, refer to the product documentation for your database. If there are any errors, or failure is indicated in your database client output, fix the reported errors and retry this step.
5. If there was an `upgradeTablespaces`*`schema_version`*`.sql` file generated for the version that you are migrating from, run it to upgrade the table space. For information about how to run an SQL script on your database, refer to the product documentation for your database. If there are any errors, or failure is indicated in your database client output, fix the reported errors and retry this step.
6. Run the `upgradeSchema`*`schema_version`*`.sql` script for the version that you are migrating from. If there are any errors, or failure is indicated in your database client output, fix the reported errors and retry this step.

   Note: When the server is started for the first time after a schema upgrade, one of the following messages is written to the SystemOut.log file:

```
CWWBB0613I: Database migration: completed successfully 700/1 to 700/0.
CWWBB0615E: Database migration failed 700/1 to 700/0.
```

The value after the“/” character is a binary flag that is reset to zero after successful migration, it is not part of the product version number. If the database migration failed, check the log file for other failure messages, and fix any problems before trying to restart the server again.

7. 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 3c on page 40, 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 database data:

If you are migrating from version 6.1.x, or 6.0.2.x, after migrating 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. Do not perform this data migration if you are migrating from version 6.2.

Before you begin

Refer to the following Technote for the most up-to-date information about performing the data migration: Technote 1327385.

Procedure

1. If you are using DB2 for z/OS perform the following.
   a. 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. If the Business Process Choreographer configuration that you are migrating is on cluster, make sure that you have manually run the BPMMigrateCluster tool for this cluster.

3. On the node where you will run the database migration script, run the syncNode command to synchronize the node with the deployment manager.
4. Run the database migration script as described in Business Process Choreographer data migration script.

   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.
5. 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.

   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 4 on page 42. 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.

6. 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 41.

Results

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

Migrating the Business Space database schema:

After migrating the server from version 6.1.2 or version 6.2.0.x, you must manually migrate the Business Space database to a new database schema before you start the version 7.0 server.

Before you begin

See the "Migrating stand-alone servers" on page 30 and "Migrating a deployment manager" on page 33 topics.

About this task

This procedure supports the Business Space database schema migration for the following database types:

Database type	Directory name
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

Database type	Directory name
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 In version 6.1.0 of WebSphere Process Server, the Cloudscape database was replaced with Derby. Under most circumstances, the profile migration tools automatically migrate Cloudscape databases to Derby. Exceptions are handled by the Migrating IBM Cloudscape or Apache Derby databases topic.

Procedure

For WebSphere Process Server for z/OS and WebSphere Enterprise Service Bus for z/OS: If you are configuring DB2 for z/OS, you can use the `createDB.sh` script to configure the Business Space database tables with all other database objects in one database. For more information, see "Creating DB2 database objects using the createDB.sh script" in the WebSphere Process Server for z/OS documentation. Alternatively, you can run the `createTables_BusinessSpace.sql`, `createTablespace.sql`, and `createTable.sql` scripts using a DB2 for z/OS utility such as SPUFI or DSNTEP2. For more information, see "Creating the DB2 database and storage groups using DButility.sh, SPUFI, or DSNTEP2" in the WebSphere Process Server for z/OS and the WebSphere Enterprise Service Bus for z/OS documentation.

Results

The database schema has been migrated, and is ready for use by Business Space version 7.0.

What to do next

- Update the endpoints for widgets that you want to be available in Business Space.
- Set up security for Business Space and the widgets that your team is using.

Migrating the Business Space database data:

After migrating the Business Space database schema, you must migrate the Business Space database data.

Before you begin

Migrate the Business Space database schema.

Note: When migrating Business Space data, the personalized information that is migrated for every Business Space user is limited to 10 of the most recently viewed pages and 60 of the most recently adjusted widgets.

Procedure

1. **Copy the widget definition files.**

   During profile migration, the version 6.2.0 and version 6.1.2 widget definition files are automatically copied to the following directory on the version 7.0 target server: `*profile_root*/BusinessSpace/datamigration/widgets`. However, the version 7.0 widget definition files and any version 6.2.0 or version 6.1.2 custom widget definition files must be copied to this directory manually.

   Depending on your environment, use one of the following procedures:

   - For a stand-alone or non-clustered managed-node environment, copy the widget files to the target profile.
   - For a clustered Business Space environment, copy the widget files on all the profiles participating in the cluster.

   To copy the widget definition files, use the following procedure.

   a. Copy all of the non-custom Business Space version 7.0 widget definition files into the *profile_root*/BusinessSpace/datamigration/widgets directory. These files can be found by searching for file names containing either *iwidget.xml* or *iWidget.xml* in the version 7.0 *profile_root*/installedApps directory.

      Note: If a warning about overwriting files appears, accept it. This means you are overwriting the non-custom version 6.2.0 or version 6.1.2 widget definition files that were copied automatically during profile migration with the new version 7.0 non-custom widget definition files.

   b. If you have custom widgets from version 6.2.0 or version 6.1.2, you must copy all of the custom widget definition files to the version 7.0 installation of Business Space before migrating the Business Space data. To do this, copy all of the custom widget definition files from the earlier versions of Business Space into the *profile_root*/BusinessSpace/datamigration/widgets directory.

2. **Start the server in the target environment.** Depending on your environment, use one of the following procedures:

   - For a stand-alone environment, start the target server.

     Start the migration target server using the `startServer` command from the *profile_root*/bin directory of the migration target server or from the target profile's First steps console.

     For more information about the `startServer` command, see the startServer command topic on the WebSphere Application Server, Version 7.0 information center.

   - For a network deployment environment, use the following procedure.

     Important: Perform the procedure using one of the following methods, depending on how the network deployment environment is configured:

     – If the Business Space database being updated belongs to a non-clustered managed node where Business Space is configured, start the node agent and the server on the node.

     – If the Business Space database being updated belongs to a clustered environment, select a node that is participating in the cluster and start the node agent and server on it.

       Note: For a Business Space clustered environment, only one node participating in the cluster needs to be started.

   a. **Start the migration target node agent.**

Start the migration target node agent using the `startNode` command from the *profile_root*/bin directory of the migration target server or from the profile's First steps console.

For more information about the `startNode` command, see the startNode command topic on the WebSphere Application Server, Version 7.0 information center.

b. **Start the migration target servers.**

Start the migration target server using the `startServer` command from the *profile_root*/bin directory of the migration target server or from the profile's First steps console.

For more information about the `startServer` command, see the startServer command topic on the WebSphere Application Server, Version 7.0 information center.

3. **Migrate the Business Space data.**

   On the node for which the target server was started in the previous step, run the `migrateBSpaceData` script to migrate Business Space version 6.1.2 or version 6.2.0 data to Business Space version 7.0.

   The script is located in the following directory: *install_root*/BusinessSpace/scripts/. For more information about the `migrateBSpaceData` script, see migrateBSpaceData command-line utility.

4. Optional: **Migrate the widget catalog for custom widgets.**

   If you have custom widgets and you are migrating a network deployment environment, you must run the `updateBSpaceWidgets` command on the deployment manager profile to populate the migrated widget catalog of the custom widgets that were generated in XML format under the following folder: *profile_root*/BusinessSpace/datamigration/catalog. Launch the `updateBSpaceWidgets` command from the *profile_root*\bin directory of the deployment manager profile.

   Example

   ```
   wsadmin>$AdminTask updateBusinessSpaceWidgets {-clusterName cluster_name
   -catalogs profile_root/BusinessSpace/datamigration/catalog }
   ```

   Note: Catalog files are generated only if you have custom widgets.

   For more information about the `updateBSpaceWidgets` command, see updateBusinessSpaceWidgets command.

5. **Migrate the widget endpoints for both product and custom widgets.**

   If you are migrating a network deployment environment, run the `updateBSpaceWidgets` command on the deployment manager profile to populate the migrated widget endpoints for both product and custom widgets that were generated in XML format under the following folder: *profile_root*/BusinessSpace/datamigration/endpoints. Launch the `updateBSpaceWidgets` command from the *profile_root*\bin directory of the deployment manager profile.

   For more information about the `updateBSpaceWidgets` command, see updateBusinessSpaceWidgets command.

   Example

   ```
   wsadmin>$AdminTask updateBusinessSpaceWidgets {-clusterName cluster_name
   -endpoints profile_root/BusinessSpace/datamigration/endpoint }
   ```

6. **Stop the target server.** Depending on your environment, use one of the following procedures:

   - For a stand-alone environment, stop the target server.

Stop the migration target server using the `stopServer` command from the *`profile_root`*`/bin` directory on the migration target system.

Note:

– If the profile has security enabled the user name provided must be a member of the operator or administrator role.

– 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 server.

– If the profile does not have security enabled the `-username` and `-password` parameters are not necessary.

For more information about the `stopServer` command, see the stopServer command topic on the WebSphere Application Server, Version 7.0 information center.

- For a network deployment environment, stop the servers in the target cluster that were started in Step 2.

  Repeat this step for each server in the cluster.

  Stop the migration target server using the `stopServer` command from the *`profile_root`*`/bin` directory on the migration source target.

  If the profile has security enabled the user name provided must be a member of the operator or administrator role.

  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 server.

  If the profile does not have security enabled the `-username` and `-password` parameters are unnecessary.

  For more information about the `stopServer` command, see the stopServer command topic on the WebSphere Application Server, Version 7.0 information center.

Results

The Business Space database data is migrated to the Business Space version 7.0.

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 BPMMigrateProfile command and the BPMMigrateCluster command.
   a. Check the file *`backupDirectory`*`/logs/BPMMigrateProfile.`*`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 BPMSnapshotSourceProfile or BPMMigrateProfile commands.

  Note: *profileName* is the name of the new profile you created in version 7.0 of WebSphere Process Server.

b. Check the file `*backupDirectory*/logs/BPMMigrateCluster.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. Verify the Common database upgrade. If the Common database upgrade was not performed manually because the user configured forWebSphere Process Server has all the necessary permissions, check that the database was upgraded successfully during deployment manager startup.
   a. Navigate to the profile directory for the deployment manager. Typically this is `install_root/profiles/<profile name>` .
   b. Navigate to the *logs* folder and check the `SystemOut.log` file. Look for the messages `The Common Database Schema upgrade is started` and `CWLDB0003I: WebSphere Process Server Schema version was updated to "7.0.0.0" successfully.`
3. Check the server log files. For example, check the logs for a stand-alone server in the `/WebSphere/Server1/AppServer/profiles/default/logs` directory.
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 corner 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 “Runtime migration troubleshooting” on page 66 for troubleshooting information.

Rolling back your environment

After migrating to a WebSphere Process Server version 7.0 environment, you can roll back to the version you migrated from, which can be a version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 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.

Rolling back a deployment cell:

You can use the **restoreConfig** and **wsadmin** commands to roll back a migrated WebSphere Process Server version 7.0 deployment cell to version 6.2.0, 6.1.2, 6.1.0, or 6.0.2. 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 6.2.0, 6.1.2, 6.1.0, or 6.0.2 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 6.2.0, 6.1.2, 6.1.0, or 6.0.2 deployment manager configuration.

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

     See the backupConfig command on the WebSphere Application Server information center.
   - Run the **backupConfig** command or your own preferred utility to back up the version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 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 on the WebSphere Application Server information center.
3. Migrate the deployment cell.

Procedure

1. Stop all of the servers that are currently running in the WebSphere Process Server version 7.0 environment.
2. If you chose to disable the previous deployment manager when you migrated to the version 7.0 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 6.2.0, 6.1.2, 6.1.0, or 6.0.2 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 on the WebSphere Application Server information center.

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

   Use the following parameters:

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

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

   1) Go to the following directory:

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

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

   2) If you see a `serverindex.xml_disabled` file in in this directory, 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 6.2.0, 6.1.2, 6.1.0, or 6.0.2 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 on the WebSphere Application Server information center.

   b. If you did not back up your previous managed node configuration, use the **wsadmin** command to run the `migrationDisablementReversal.jacl` script from the version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 *profile_root*`/bin` directory of the managed node.

   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`.

c. If you did not back up your previous managed node configuration, use the **wsadmin** command to run the `migrationDisablementReversal.jacl` script from the version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 *`install_root`*`/bin` directory of the managed node.

   Use the following parameters:

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

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

   1) Go to the following directory:

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

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

   2) If you see a `serverindex.xml_disabled` file in in this directory, 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 7.0 deployment manager was running.

   See syncNode command on the WebSphere Application Server information center.

5. If you chose to keep the installed applications in the same location as the prior release during migration to version 7.0 and any of the version 7.0 applications are not compatible with the prior release, install applications that are compatible.

6. Delete the version 7.0 profiles.

   See Deleting a profile on the WebSphere Application Server information center.

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

8. Start the rolled-back deployment manager and its managed nodes in the version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 environment.

9. Enable synchronization for all the nodes if it was disabled previously. To do this, use the following procedure.

   a. From the WebSphere Application Server administrative console, select **System administration** → **Node agents**.

   b. Click the node agent for the node.

   c. Click **File synchronization service**.

   d. Select **Enable service at server startup**, **Automatic synchronization** and **Startup synchronization**.

   e. Click **Apply**, then click **OK** to save the configuration changes.

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.

Rolling back a managed node:

You can use the **restoreConfig** and **wsadmin** commands to roll back a migrated WebSphere Process Server version 7.0 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 6.2.0, 6.1.2, 6.1.0, or 6.0.2 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 6.2.0, 6.1.2, 6.1.0, or 6.0.2 deployment manager configuration..

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

     See the backupConfig command in the WebSphere Application Server information center.
   - Run the **backupConfig** command or your own preferred utility to back up the version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 managed node configuration.

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

     See the backupConfig command in the WebSphere Application Server information center.
3. Migrate the managed node.

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

Important: If you do not have a backup copy of your version 7.0 deployment manager configuration as it was before you migrated the version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 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 49.

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 7.0 environment.
3. Restore your previous configuration.
   a. Run the **restoreConfig** command or your own preferred utility to restore the version 7.0 deployment manager configuration.

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

      See the restoreConfig command in the WebSphere Application Server Information Center.
   b. Perform one of the following actions to restore the version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 configuration for the managed node.
      - Run the **restoreConfig** command or your own preferred utility to restore the version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 configuration.

        See restoreConfig command in the WebSphere Application Server Information Center.
      - Use the **wsadmin** command to run the `migrationDisablementReversal.jacl` script from the version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 *`install_root`*`/bin` directory of the managed node that you need to roll back from version 7.0.

        Use the following parameters:

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

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

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

           where *node_name* is the name of the managed node that you want to roll back.
        2) If you see a `serverindex.xml_disabled` file in in this directory, perform the following actions:
           a) Delete or rename the `serverindex.xml` file.
           b) Rename the `serverindex.xml_disabled` file to `serverindex.xml`.
4. Start the version 7.0 deployment manager.
5. Synchronize the managed node.

   See Synchronizing nodes with the wsadmin tool in the WebSphere Application Server information center.
6. If you chose to keep the installed applications in the same location as the prior release during migration to version 7.0 and any of the version 7.0 applications are not compatible with the prior release, install applications that are compatible.
7. Start the rolled-back managed node in the version 7.0 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.

Postmigration tasks

Postmigration tasks are task you perform on WebSphere Process Server, Business Process Choreographer, and Business Space after successfully migrating to version 7.0.

Postmigration tasks for WebSphere Process Server

After migration, you might need to check some configuration settings, or further configure the version 7.0 server.

Before you begin

Ensure that you 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 6.2.0, 6.1.2, 6.1.0, or 6.0.2, and make sure that version 7.0 security is set appropriately.
- Check the `BPMMigrateProfile.`*`profile_name`*`.`*`timestamp`*`.log` file in the `logs` directory for details about any JSP objects that the migration tools did not migrate.

  If version 7.0 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.
- After migrating from version 6.2.0.x to version 7.0, you check your WebSphere Adapter properties to ensure that they are properly configured for the new installation location. Some adapter properties might need to be altered during migration in a way that would be unknown to an automated migration.
- If you had uninstalled applications using Websphere Adapters version 6.0.2.x before migration as mentioned in Runtime premigration checklist, you can update the applications using Websphere Integration Developer to use Websphere Adapter version 7.0.x and install them in 7.0.x environment. You can install these updated applications using administrative console.
- Optional: After migrating to version 7.0, be aware that the default value for the target significance property has changed from version 6.2.0, 6.1.2, 6.1.0, or 6.0.2. In version 7.0, default value was changed from `targetSignificance=prefered` to `targetSignificance=required`. The new default value is set in the JMS activation specifications and connection factories that are part of the WebSphere Process Server configuration.

  You must determine whether to change the target significance value in the migrated environment (version 6.2.0, 6.1.2, 6.1.0, or 6.0.2).
- If the version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 environment that you have migrated includes a Business Process Choreographer Observer application, and your post postmigration work involves moving Business Process Choreographer Explorer

to a new deployment target, the Business Process Choreographer Observer application is not moved along with the Business Process Choreographer Explorer. In such a scenario, you would be required to merge the pre-62 Business Process Choreographer Observer application with the moved Business Process Choreographer Explorer configuration and then delete the old Business Process Choreographer Observer application.

Alternatively, you could merge the migrated Observer with the Explorer that is to be moved BEFORE moving the Explorer. Then the combined Explorer (now with the Observer reporting function) would be moved.

- After migrating to version 7.0, 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**.

The migration tools convert appropriate command-line parameters to Java virtual machine settings in the process server definition. Most settings are mapped directly, but some settings are not migrated because their roles differ in WebSphere Application Server version 7.0. In such cases, the configuration settings might not exist, they might have different meanings, or they might have different scopes. See the following topics in the WebSphere Application Server version 7.0 information center for more information about changing the process definition settings or JVM settings:

- Process definition settings
- Java virtual machine settings

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, if necessary, 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 Server version 7.0 in production.

Procedure

1. If you used people assignment before migrating to version 7.0, you must perform the following:
   a. If you have applied any changes to the default XSL transformation files (`EverybodyTransformation.xsl`, `LDAPTransformation.xsl`, `SystemTransformation.xsl`, `VMMTransformation.xsl`, and `UserRegistryTransformation.xsl`) that are located in the

install_root/ProcessChoreographer/Staff directory then you must re-apply your changes to the WebSphere Process Server version 7.0 versions of these files after migration. In a clustered environment, the transformation file must be available on the deployment manager and on each node that hosts members of the cluster where Business Process Choreographer is configured. Make sure that they all use the same version of the transformation file.

Note: Custom XSL transformation files located in the *install_root*/ ProcessChoreographer/Staff directory are 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 6.2.0, 6.1.2, 6.1.0, or 6.0.2 people directory configuration (previously known as the staff plug-in configuration).

b. If you used the substitution feature and substitution information is stored in one of the user repositories that are configured for VMM, you must add the new properties for substitutionStartDate and substitutionEndDate to your repository. The steps that you must perform depends on whether you store the substitution information in the VMM file registry or in the VMM property extension registry:

For the VMM file registry:

1) Add the substitutionStartDate and substitutionEndDate properties to the definition of the PersonAccount entity type in the file wimxmlextension.xml file.The file is located in *profile_root*/config/cells/*cellName*/wim/model. In a network deployment environment, edit the file on the deployment manager.Extend the file to include the new properties, which are highlighted in bold:

```
<wim:propertySchema nsURI="http://www.ibm.com/websphere/wim"
     dataType="STRING" multiValued="false" propertyName="isAbsent">
    <wim:applicableEntityTypeNames>PersonAccount</wim:applicableEntityTypeNames>
</wim:propertySchema>

<wim:propertySchema nsURI="http://www.ibm.com/websphere/wim"
     dataType="STRING" multiValued="true" propertyName="substitutes">
     <wim:applicableEntityTypeNames>PersonAccount</wim:applicableEntityTypeNames>
</wim:propertySchema>

<wim:propertySchema nsURI="http://www.ibm.com/websphere/wim"
     dataType="STRING" multiValued="false" propertyName="substitutionStartDate">
    <wim:applicableEntityTypeNames>PersonAccount</wim:applicableEntityTypeNames>
</wim:propertySchema>

<wim:propertySchema nsURI="http://www.ibm.com/websphere/wim"
     dataType="STRING" multiValued="false" propertyName="substitutionEndDate">
     <wim:applicableEntityTypeNames>PersonAccount</wim:applicableEntityTypeNames>
</wim:propertySchema>
```

2) The changes will become effective when the servers are restarted. In a network deployment environment the deployment manager must also be restarted.

For the VMM property extension registry:

1) Check that the substitution properties isAbsent and substitutes are defined for the property extension repository. If they were not defined before migration, no substitution information was stored in the VMM property extension repository, and this migration step is not required.

Change to the directory *install_root*/bin and enter the following commands in either local mode or in connected mode. In a network deployment environment, enter the commands on the deployment manager.

```
wsadmin -username admin -password adminPassWord
$AdminTask listIdMgrPropertyExtensions
```

2) Add the new properties substitutionStartDate and substitutionEndDate to the property extension repository configuration by entering the following commands:

```
$AdminTask addIdMgrPropertyToEntityTypes
   {-name substitutionStartDate
    -dataType String
    -isMultiValued false
    -entityTypeNames PersonAccount
    -repositoryIds LA}

$AdminTask addIdMgrPropertyToEntityTypes
   {-name substitutionEndDate
    -dataType String
    -isMultiValued false
    -entityTypeNames PersonAccount
    -repositoryIds LA}
```

3) The changes will become effective when the servers are restarted. In a network deployment environment the deployment manager must also be restarted.

4) Verify that the new properties have been added to the property extension repository configuration by entering the following command:

```
$AdminTask listIdMgrPropertyExtensions
```

2. Configure the REST API endpoints for the Business Flow Manager and Human Task Manager, update all references, and map Web modules to a Web server.

   a. 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:

      1) To change the endpoint for the Business Flow Manager, click either **Servers** → **Clusters** → **WebSphere application server clusters** → *cluster_name* or **Servers** → **Server Types** → **WebSphere application servers** → *server_name*, then under **Business Integration**, expand **Business Process Choreographer**, and click **Business Flow Manager**, and under **Additional Properties** click **REST Service Endpoint**.

      2) To change the endpoint for the Human Task Manager, click either **Servers** → **Clusters** → **WebSphere application server clusters** → *cluster_name* or **Servers** → **Server Types** → **WebSphere application servers** → *server_name*, then under **Business Integration**, expand **Business Process Choreographer**, and click **Human Task Manager**, and under **Additional Properties** click **REST Service Endpoint**.

   b. If you migrated from version 6.2 or later, and you still use the bpcEndpoints.xml file, the endpoint configuration is not migrated automatically, so you must use the administrative console to make sure that the references to the REST APIs for Business Space are correct. Note that since version 6.2, you should configure the Business Process Choreographer REST API endpoints for Business Space using the administrative console rather than using the bpcEndpoints.xml file. To check or change the Business Process Choreographer REST API endpoints for Business Space:

1) To change the endpoint for the Business Flow Manager, click either **Servers** → **Clusters** → **WebSphere application server clusters** → *cluster_name* or **Servers** → **Server Types** → **WebSphere application servers** → *server_name*, then under **Business Integration**, expand **Business Process Choreographer**, and click **Business Flow Manager**, and under **Additional Properties** click **REST Service Endpoint**.
2) To change the endpoint for the Human Task Manager, click either **Servers** → **Clusters** → **WebSphere application server clusters** → *cluster_name* or **Servers** → **Server Types** → **WebSphere application servers** → *server_name*, then under **Business Integration**, expand **Business Process Choreographer**, and click **Human Task Manager**, and under **Additional Properties** click **REST Service Endpoint**.
3) To register these endpoints with Business Space, click either **Servers** → **Clusters** → **WebSphere application server clusters** → *cluster_name* or **Servers** → **Server Types** → **WebSphere application servers** → *server_name*, then under **Business Integration**, expand **Business Space Configuration**, then under **Additional Properties** click **REST service endpoint registration**, and make sure that the correct **Service Endpoint Target** for the Business Flow Manager service and Human Task Manager service are selected.

c. The REST APIs were configured during 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.
   1) To change the context root:
      a) In the administrative console click **Applications** → **Application Types** → **WebSphere 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** → **Application Types** → **WebSphere 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.
   2) To change the endpoint references for Business Process Choreographer Explorer, click either **Servers** → **Clusters** → **WebSphere application server clusters** → *cluster_name* or **Servers** → **Server Types** → **WebSphere application servers** → *server_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.
   3) To change the endpoint references for Business Space:
      a) To change the endpoint for the Business Flow Manager, click either **Servers** → **Clusters** → **WebSphere application server clusters** → *cluster_name* or **Servers** → **Server Types** → **WebSphere application servers** → *server_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** → **Clusters** → **WebSphere application server clusters** → *cluster_name* or **Servers** → **Server Types** → **WebSphere application servers** → *server_name*, then under **Business Integration**, expand **Business Process Choreographer**, and click **Human Task Manager**, and under **Additional Properties** click **REST Service Endpoint**.

d. The JAX Web Services APIs were configured during migration. You might want to map the Web modules to a Web server and change the context root for the JAX Web Services APIs Web modules.

To change the context root:

1) In the administrative console click **Applications** → **Application Types** → **WebSphere 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.

2) Make sure that the context root for the Web module BFMJAXWSAPI is correct and unique.

3) In the administrative console click **Applications** → **Application Types** → **WebSphere enterprise applications** → **TaskContainer_*suffix*** → **Context Root for Web Modules**

4) Make sure that the context root for the Web module HTMJAXWSAPI is correct and unique.

3. If you performed the “minimum downtime” scenario to migrate a cluster, you must run the bpeupgrade.jacl script to deploy the new versions of the predefined human tasks and to ensure that the new Business Process Choreographer JAX Web Services APIs are added.

CAUTION:
Do not try to use the administrative console to update the predefined human task applications.

a. Stop the deployment manager.

b. On the deployment manager, change to the directory where the bpeupgrade.jacl script is located, and run the script. Change to the directory `install_root/ProcessChoreographer/config` and enter the command:

```
../../bin/wsadmin.sh -conntype NONE -profileName profileName
                     -f bpeupgrade.jacl -cluster clusterName
```

c. Start the deployment manager.

d. Synchronize the configuration changes with the nodes and restart the cluster members.

4. When there are no more instances of old versions of the predefined human tasks running, remove them.

Note: Because there might still be running instances of the old predefined human task applications, the old predefined human task applications are not uninstalled during migration. This means that after migration, both the new and old versions of the predefined human task applications are in your system. The version numbering indicates when the application was last updated, which can look older than the current release, but that just means that it has not changed.

a. Make sure that all old instances have been deleted.

b. In the administrative console, click **Applications** → **Application Types** → **WebSphere enterprise applications**

c. If there are multiple versions of any of the following applications, select the older applications and click **Uninstall**.

   - HTM_PredefinedTasks_V*nnn_scope*.ear
   - HTM_PredefinedTaskMsg_V*nnn_scope*.ear

   where

   nnn is the version number when the application was last updated, for example `620`. If the newest version of these applications looks older than the current release, it just means that it has not changed. The important thing is to only delete the oldest if there are more than one version of the two applications.

   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: If you migrated from version 6.1.x or 6.0.2.x, you can release the extra storage space used by the work item data migration by deleting the table WI_ASSOC_OID_T from the database.

6. Optional: If you migrated from version 6.1.x, or 6.0.2, and you use DB2 for 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 migrated from version 6.0.2 or 6.1.x, and 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 written a client for version 6.0.2 that used 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 Java EE roles BPEAPIUser and TaskAPIUser are set to the value Everyone, which provides compatibility with earlier versions by maintaining the version 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** → **Application Types** → **WebSphere 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 Java EE 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 have written an application that uses the Business Process Choreographer EJB APIs and you packaged one or both of the `bpe137650.jar` and `task137650.jar` files that contain the EJB stubs with your application then remove these utility JAR files.

11. If you modified the `faces-config-beans.xml` configuration file to specify thresholds for the queries for the Business Process Choreographer Explorer before migrating to version 7.0, 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.

12. Optional: Change the business process navigation mode to the new default. From version 7.0, the default navigation mode for business processes uses the work-manager. Before version 7.0, the default navigation mode used JMS messaging. Because the navigation mode is not changed during migration, if you want to improve performance by using the work-manager-based navigation, you must select it manually, as described in the related link.

13. Optional: Change the database retention behavior for iterated inline human tasks. Before version 7.0, inline human tasks that were processed as part of multiple “while” loops or “repeat-until” loops have been kept in the database by default. The new default behavior, starting with version 7.0, is that if “while” loops or “repeat-until” loops iterate multiple times, the inline human tasks that were processed in previous iterations are deleted from the database.

    If you want to maintain the previous behavior for both types of loop in migrated environments, you must add a new custom property manually. In the administrative console, click **Servers** → **Clusters** → **WebSphere application server clusters** → ***cluster_name*** or **Servers** → **Server Types** → **WebSphere application servers** → ***server_name***, then under **Business Integration**, expand **Business Process Choreographer**, and click **Business Flow Manager** → **Custom Properties**.. Then add a property named `InlineHumanTasks.KeepOverMultipleWhileLoopIterations` with the value `true`. When you no longer want the old behavior, you must delete the custom property.

14. If you want to use WebSphere Business Monitor to monitor Service Component Architecture (SCA) events, you must set a custom property to enable SCA events.

    a. In the administrative console, click **Servers** → **Clusters** → **WebSphere application server clusters** → ***cluster_name*** or **Servers** → **Server Types** →

WebSphere application servers → *server_name*, then under **Business Integration**, expand **Business Process Choreographer**, and click **Business Flow Manager** → **Custom Properties**.

b. Click **New** to add a new custom property.

c. Enter the name `Compat.SCAMonitoringForBFMAPI` and the value `true`.

d. Save the changes. The setting will be activated the next time that you restart the server or cluster where Business Process Choreographer is configured.

Related information

Uninstalling business process and human task applications, using the administrative console

Uninstalling business process and human task applications, using administrative commands

Improving the performance of business process navigation

Postmigration tasks for Business Space powered by WebSphere

After migrating WebSphere Process Server from version 6.1.2 orversion 6.2.0 to version 7.0, 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 or version 6.2.0 and you have Business Space configured, you must perform the following steps after migration before you can use Business Space.

Procedure

1. If you had custom widgets in Business Space version 6.1.2 or version 6.2.0, you must perform some manual steps to make these widgets operational in Business Space version 7.0. For more details, see the Business Space Development Guide.
2. If you have Business Space widgets enabled for remote endpoint, you must migrate them manually. To do this, use the following procedure.
   a. Copy the catalog registry files for specific BPM product widgets from the source to the target installation.
   b. Copy the endpoint files for specific BPM product widgets from the source to the target.
   c. Modify the endpoint files for specific BPM product widgets in the target to update the TNS URL. For instructions, see Enabling Business Space widgets for cross-cell environments.
   d. Register the catalog and endpoint information for the specific BPM product widgets on Business Space in the target installation using the `updateBusinessSpaceWidgets` command. For more information, see updateBusinessSpaceWidgets command.

Results

You can use Business Space version 7.0.

Note: If you have used Business Space version 6.1.2, you must clear your browser cache before using Business Space version 7.0. This will help you avoid inadvertent, continued use of code and images from Business Space version 6.1.2.

Runtime migration tools reference

Use the runtime migration tools to migrate topology configuration, applications, and databases to WebSphere Process Server version 7.0.

The runtime migration tools required to perform a version-to-version migration fall into the following categories:

“BPM profile command-line tools”

“BPM database upgrade command-line utilities” on page 64

“WebSphere Application Server command-line utilities” on page 64

BPM profile command-line tools

BPMSnapshotSourceProfile

The BPMSnapshotSourceProfile command copies the configuration files in the source profile to a snapshot directory that will serve as the source of the profile migration.

For more information about the `BPMSnapshotSourceProfile` command, see the BPMSnapshotSourceProfile command-line utility topic.

BPMCreateTargetProfile

The BPMCreateTargetProfile command creates a target migration profile using some of the base configuration information that was backed up using the BPMSnapshotSourceProfile command.

For more information about the `BPMCreateTargetProfile` command, see the BPMCreateTargetProfile command topic.

BPMMigrateProfile

The BPMMigrateProfile command migrates a source profile from the snapshot directory to a target profile.

For more information about the `BPMMigrateProfile` command, see the BPMMigrateProfile command topic.

BPMMigrateCluster

The `BPMMigrateCluster` command migrates cluster scoped application and configuration information.

For more information about the `BPMMigrateCluster` command, see the BPMMigrateCluster command topic.

BPMMigrationStatus

The BPMMigrationStatus command displays the status of the migrations that have been executed on the system.

For more information about the `BPMMigrationStatus` command, see the BPMMigrationStatus command topic.

BPMCreateRemoteMigrationUtilities

The BPMCreateRemoteMigrationUtilities command creates an archive file containing all the commands and their prerequisites that need to be invoked on the system containing the source profile to be migrated.

For more information about the `BPMCreateRemoteMigrationUtilities` command, see the BPMCreateRemoteMigrationUtilities command topic.

installBRManager

The installBRManager command migrates the Business Rules Manager.

For more information about the `installBRManager` command, see the installBRManager command topic.

BPM database upgrade command-line utilities

migrateDB (Business Process Choreographer)

If you are migrating from version 6.1.x, or 6.0.2.x, use the migrateDB.py script to migrate the runtime data in the Business Process Choreographer database to the new schema. The new schema provides better query performance for business processes and human tasks.

For more information about the `migrateDB` command, see the Business Process Choreographer data migration script topic.

migrateSchema (Business Space)

Use the migrateSchema command-line utility to migrate the Business Space database schema.

For more information about the `migrateSchema` command, see the migrateSchema command-line utility for the Business Space database topic.

updateBspaceData (Business Space)

Use the `migrateBSpaceData` command-line utility to migrate the Business Space data.

For more information about the `migrateData` command, see the migrateBSpaceData command-line utility topic.

upgradeSchema (Common Database)

Use the upgradeSchema command-line utility to upgrade the Common database schema.

For more information about the `upgradeSchema` command, see the upgradeSchema command-line utility for the Common database topic.

upgradeDB.sh script

Use the upgradeDB.sh script to upgrade the WebSphere Process Server databases: the Common database, the Business Process Choreographer database, and the Business Space database.

For more information about the `upgradeDB` command, see the upgradeDB.sh script topic.

WebSphere Application Server command-line utilities

backupConfig

The `backupConfig` command is a simple utility to back up the configuration of your node to a file.

For more information about the `backupConfig` command, see the backupConfig command topic on the WebSphere Application Server information center.

convertScriptCompatibility

The `convertScriptCompatibility` command is used by administrators to convert their configurations from a mode that supports backward compatibility of WebSphere Application Server Version 5.1.x or Version 6.0.x administration scripts to a mode that is fully in the Version 7.0 configuration model.

For more information about the `convertScriptCompatibility` command, see the convertScriptCompatibility command topic on the WebSphere Application Server information center.

migrationDisablementReversal

If you need to roll back a deployment cell or managed node, use the `wsadmin` command to run the `migrationDisablementReversal.jacl` script.

For more information about the `migrationDisablementReversal.jacl` script, see the Rolling back a Network Deployment cell topic on the WebSphere Application Server information center.

restoreConfig

Use the `restoreConfig` command to restore the configuration of your node after backing up the configuration using the `backupConfig` command.

For more information about the `restoreConfig` command, see the restoreConfig command topic on the WebSphere Application Server information center.

startManager

Use the `startManager` command to manipulate a deployment manager with scripting.

For more information about the `startManager` command, see the startManager command topic on the WebSphere Application Server information center.

startNode

The `startNode` command reads the configuration file for the node agent process and constructs a launch command.

For more information about the `startNode` command, see the startNode command topic on the WebSphere Application Server information center.

startServer

The `startServer` command reads the configuration file for the specified server process and starts that server process.

For more information about the `startServer` command, see the startServer command topic on the WebSphere Application Server information center.

stopManager

The `stopManager` command reads the configuration file for the Network Deployment manager process.

For more information about the `stopManager` command, see the stopManager command topic on the WebSphere Application Server information center.

stopNode

The `stopNode` command reads the configuration file for the Network Deployment node agent process and sends a Java Management Extensions (JMX) command telling the node agent to shut down.

For more information about the `stopNode` command, see the stopNode command topic on the WebSphere Application Server information center.

stopServer

The `stopServer` command reads the configuration file for the specified server process. This command sends a Java management extensions (JMX) command to the server telling it to shut down.

For more information about the `stopServer` command, see the stopServer command topic on the WebSphere Application Server information center.

syncNode

The `syncNode` command forces a configuration synchronization to occur between the node and the deployment manager for the cell in which the node is configured.

The node agent server runs a configuration synchronization service that keeps the node configuration synchronized with the master cell configuration. If the node agent is unable to run because of a problem in the node configuration, you can use the `syncNode` command to perform a synchronization when the node agent is not running in order to force the node configuration back in sync with the cell configuration. If the node agent is running and you want to run the `syncNode` command, you must first stop the node agent.

For more information on the `syncNode` command, see the syncNode command topic on the WebSphere Application Server information center.

Runtime migration troubleshooting

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

The following sections describe specific errors and exceptions that may occur in a BPM runtime version migration and provide steps you can follow to understand and resolve these problems.

- “Application installation error” on page 67
- “Application server error” on page 67
- “BPMSnapshotSourceProfile and BPMMigrateProfile errors” on page 68
- “Business Rules Manager is not automatically migrated” on page 71
- “Communication with deployment manager error” on page 71
- “ConnectorException” on page 71
- “Exceptions: database connectivity, loading, or missing class” on page 71
- “Out of memory error” on page 72
- “Servlet error” on page 72
- “Synchronization error” on page 73

- “WebSphere Process Server client migrations” on page 73
- “WSDL validation exception” on page 73

Application installation error

If you select the option for the migration process to install the enterprise applications that exist in the version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 configuration into the new version 7.0 configuration, you might encounter some error messages during the application-installation phase of migration.

The applications that exist in the version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 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 BPMMigrateProfile 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 6.2.0, 6.1.2, 6.1.0, or 6.0.2 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 7.0 configuration using the administrative console or an install script.

Application server error

After you migrate a managed node to version 7.0, 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.

BPMSnapshotSourceProfile and BPMMigrateProfile errors

If you encounter a problem when you are migrating from a earlier version of WebSphere Process Server to version 7.0, check your log files and other available information.

- If the migration job fails before the BPMMigrateProfile step, rerun the migration job.
- If the migration job fails in the BPMMigrateProfile step, the configuration of the new version 7.0 server is partially updated so re-create (or restore from backup) the new version 7.0 server before rerunning the migration job.
- Problems occur with a managed (federated) node migration.

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

  version 6.x node agents might display as not synchronized or not available when you change the deployment manager node name in a mixed cell during migration to the version 7.0 deployment manager. Version 6.x node agents maintain a link to the version 6.x deployment manager until they are restarted; therefore, they might fail to synchronize with the new deployment manager. The discovery problem, which prevents automatic synchronization, occurs because the node agent is not yet aware of the deployment manager name change that occurred during the migration. If you experience this problem, perform these steps on the node.

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

  If you select the option for the migration process to install the enterprise applications that exist in the version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 configuration into the new version 7.0 configuration, you might encounter error messages during the application-installation phase of migration.

  The applications that exist in the version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 configuration might have incorrect deployment information—usually, invalid XML documents that were not validated sufficiently in previous WebSphere Process Server runtimes. The runtime now has an improved application-installation validation process and will fail to install these malformed EAR files. This results in a failure during the application-installation phase of BPMMigrateProfile 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 6.2.0, 6.1.2, 6.1.0, or 6.0.2 applications, and then remigrate.

  – Proceed with the migration and ignore these errors.

    1. Restart the migration job in the FINISHUP step to allow the remaining migration functions to be performed.

       Do this by adding the RESTART=FINISHUP parameter to the job card and resubmitting the job.

    2. Later, fix the problems in the applications and then manually install them in the new version 7.0 configuration using the administrative console or an install script.

- Out of space errors occur.

  The migration logs are located in *temporary_directory_location*/*nnnnn*, where *temporary_directory_location* is the value that you specified when you created the migration jobs (where the default is `/tmp/migrate`) and *nnnnn* is a unique number that was generated during the creation of your migration jobs. Normally, the space requirements for the migration logs are small. If you enable tracing, however, the log files can be quite large. Enable tracing only after problems have been found. If tracing is required, try to only enable tracing related to the step in the process that is being debugged. This will help to reduce the space requirements.

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

  ```
  TraceState=enabled
  profileTrace=disabled
  preUpGradeTrace=disabled
  postUpGradeTrace=enabled
  ```

  During migration, a backup copy of your version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 configuration is made. This backup becomes the source of the information being migrated. The default backup location is `/tmp/migrate/`*nnnnn*. This location can be changed when you create the migration jobs. Depending on the size of the node being migrated, this backup can be quite large. If your temporary space is inadequate, then you will need to relocate this backup.

- Batch job time is exceeded.

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

- Failures occur during server startup after migration.

  Review the instructions that were generated when you created the migration jobs. Verify that the JCL procedures have been copied over correctly to your PROCLIB, the RACF® definitions have been created, the version 7.0 libraries have been authorized, and, if required, your STEPLIB statements to the version 7.0 libraries have been specified. Make sure that the daemon process associated with your cell is at the appropriate level. The daemon process must be at the highest WebSphere Process Server for z/OS version level of all servers that it manages within the cell.

  After migrating to a version 7.0 cell that contains or interoperates with version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 nodes that are not at version 6.0.1.3 or later, the cluster function might fail. When starting these version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 application servers, you might see the following problems:

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

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

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

During migration, version 7.0 cluster information is distributed throughout the cell.version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 nodes that are not at version 6.0.1.3 or later fail to read this information. To avoid this problem, upgrade all version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 nodes that will be contained in or interoperating with a version 7.0 cell to version 6.0.1.3 or later before migrating your deployment managers to version 7.0.

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

Note: WebSphere Process Server provides an interactive problem control system (IPCS) verb exit to help you to format information from dumps of WebSphere Process Server processes. This verb exit was named CBDATA, which was an alias of the real module name, in version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 and earlier. In version 7.0, that alias was removed. In version 7.0 and later, therefore, you must use the real name of this verb exit, BBORDATA, instead of the alias.

If you migrate a node to version 7.0 then discover that you need to revert back to version 6.2.0, 6.1.2, 6.1.0, or 6.0.2, see “Rolling back your environment” on page 48.

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

Business Rules Manager is not automatically migrated

Problem

If a version 6.0.2 Business Rules Manager is not automatically migrated, the following exception appears when you launch Business Rules Manager:

```
java.lang.ClassNotFoundException:
com.ibm.wbiserver.brules.BusinessRuleManager
```

This happens because the business rule runtime has refactored and put this class into a new package in releases after version 6.0.2: com.ibm.wbiservers.brules.BusinessRuleManager.

Explanation

If the last migrated node is not a WebSphere Process Server profile, business rules resources and the Business Rules Manager migration script are not available. Therefore, Business Rules Manager is not automatically migrated during the migration process, as expected.

Solution

Run the Business Rules Manager migration script in a WebSphere Process Server custom node after the entire system is migrated. For more information, see installBRManager command-line utility.

Communication with deployment manager error

Sometimes the migration process can fail because of insufficient resources on the machine. If the migration fails, check the log file to see if the following message appears:

```
"MIGR0494E: An unexpected error occured during communication with the Deployment Manager,
the migration cannot continue. Resolve the error and rerun the WASPreUpgrade tool to
create a new backup directory."
```

If you see this message in the log file, check the disk space on the machine, memory and CPU utilization. If possible, stop some other processes on the machine to free up machine resources and rerun the migration command that has failed.

ConnectorException

When migrating a managed node, if you see a ConnectorException as shown below, ensure that your deployment manager is running and rerun the command.

```
MIGR0380E: The JMX connection is not established with the deployment manager node qaxs06,
using connector type of SOAP on port 8879. The WASPostMigration program is now closing. No
changes are made to the local Application Server environment.
com.ibm.websphere.management.exception.ConnectorException: ADMC0016E: The system cannot
create a SOAP connector to connect to host qaxs06 at port 8879.
com.ibm.ws.migration.utility.UpgradeException:
com.ibm.websphere.management.exception.ConnectorException: ADMC0016E: The system cannot
create a SOAP connector to connect to host qaxs06 at port 8879.
```

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 BPMSnapshotSourceProfile or BPMMigrateProfile 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: Handling certain Out of Memory conditions when migrating an earlier version of WebSphere Application Server to V6.0.2, V6.1, or 7.0.

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 the Business Rules Manager section in the “What gets migrated” on page 21 topic for more information.

Synchronization error

If synchronization fails when you migrate a managed node to version 7.0, the server might not start.

You might receive messages similar to the following when you migrate a managed node to version 7.0:

```
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 7.0 configuration level.
- The managed node that you are trying to migrate is at a version 7.0 configuration level on the deployment manager's repository (including applications).
- The managed node is not quite complete because 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 .

WebSphere Process Server client migrations

When migrating WebSphere Process Server client profiles from source version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 to a full server WebSphere Process Server version 7.0 installation, the target profile augmentation is not correct. Applications on the target profile might not work correctly. To correct the problem, use the manageprofiles command-line utility to add the augmentation for INSTALL_ROOT/profileTemplates/SCA/*.sdo template, where the "*" symbol represents "default" for standalone and "managed" for federated profiles.

WSDL validation exception

If the BPMMigrateProfile command fails with the following WSDL validation exception, it means that a WSDL file in the application that failed to install has an input element declaration that is not defined within an operation. To fix this problem, you must either define the input element declaration or remove it from the WSDL file.

WSDL validation exception

```
java.io.IOException: javax.wsdl.WSDLException: WSDLException (at /wsdl:definitions
/wsdl:import/wsdl:definitions/wsdl:input): faultCode=INVALID_WSDL: Encountered
illegal extension element '{http://schemas.xmlsoap.org/wsdl/}input' in the context
of a 'javax.wsdl.Definition'. Extension elements must be in a namespace other than
```

```
WSDL's.
javax.wsdl.WSDLException: WSDLException (at /wsdl:definitions/wsdl:import/wsdl:
definitions/wsdl:input): faultCode=INVALID_WSDL: Encountered illegal extension
element '{http://schemas.xmlsoap.org/wsdl/}input' in the context of a 'javax.wsdl.
Definition'. Extension elements must be in a namespace other than WSDL's.
```

How to fix the problem

Use the following procedure to fix the problem.

1. Locate the WSDL file in the application that failed to install. The WSDL file that is failing in validation has an input element declaration that is not defined within an operation.

   Sample of a failed WSDL file

   Note: The declaration for `getLastSellPriceRequest` is not defined under the `wsdl:operation` declaration.

   ```
   wsdl:portType name="EnrollIntf"
   wsdl:operation name="Enrollment"
   wsdl:input message="tns:EnrollmentRequestMsg" name="EnrollmentRequest"/
   wsdl:output message="tns:EnrollmentResponseMsg" name="EnrollmentResponse"/
   /wsdl:operation
   /wsdl:portType
   ```

   ```
   wsdl:input name="getLastSellPriceRequest"
   wsdlsoap:header message="tns:EnrollmentRequest" part="soap_header" use="literal"/
   wsdlsoap:body parts="EnrollReq" use="literal"/
   /wsdl:input
   ```
2. Make the appropriate change to the input declaration, depending on whether the input declaration file is needed or not.
   - If the input declaration is needed, move it under the operation that uses it.
   - If the input declaration is not needed, remove it from the WSDL file.
3. Update the application in source environment.
4. Verify that the application works in source environment.
5. Perform the migration steps again starting with the BPMSnapshotSourceProfile command or the BPM profile migration wizard.

Deprecated features

This section summarizes deprecated features in the following product offerings; WebSphere Process Server version 7.0, version 6.2.0, version 6.1.2, version 6.1.0, version 6.0.2, version 6.0.1, and version 6.0.

Deprecation list

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

- “Deprecated features in WebSphere Process Server version 7.0” on page 75
- “Deprecated features in WebSphere Process Server version 6.2” on page 79
- “Deprecated features in WebSphere Process Server version 6.1.2” on page 80
- “Deprecated features in WebSphere Process server version 6.1” on page 80
- “Deprecated features in WebSphere Process Server version 6.0.2” on page 84
- “Deprecated features in WebSphere Process Server version 6.0.1” on page 86
- “Deprecated features in WebSphere Process Server version 6.0” on page 86

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 7.0

Command-line utilities for version-to-version migration

The following command-line utilities for version-to-version migration are deprecated.

The deprecated command-line utilities have been replaced by new business process management command-line utilities, as outlined in the following table.

Table 2. Deprecated command-line utilities for version-to-version migration

Deprecated command-line utility	Replacement command-line utility
WBIPreUpgrade	*install_root*/bin/BPMSnapshotSourceProfile
WBIPostUpgrade	• *install_root*/bin/BPMCreateTargetProfile • *install_root*/bin/BPMMigrateProfile
WBIProfileUpdate.ant	*install_root*/bin/BPMMigrateCluster
install_root/bin/wbi_migration/ wbi_migration	*install_root*/bin/BPMMigrate

Business Process Choreographer widgets

These widgets are deprecated:

- My Tasks
- Available Tasks
- Tasks I Created
- Create Tasks

Recommended migration action:

Use the new Tasks List widget, after performing these steps for all pages with the deprecated widgets:

1. Check and capture in the Widget Wiring Editor any wiring from and to the deprecated widget.
2. Check and capture your specific configuration options for this widget.
3. Delete the widget from the page.
4. Add the Tasks List widget at the same position onto the page.
5. Configure the widget to match the configuration of the deprecated widget Make sure that you select those properties visible that you would like to filter for or sort by.
6. Configure the scenario context to match the deprecated widget:
   - Work with tasks assigned to you to replace the My Tasks widget.
   - Assess tasks available for you to replace the Available Tasks widget.
   - Check status of initiated tasks, services, and processes to replace the Tasks I Created widget.
7. Add explicit wires matching the previous wiring.
8. Add explicit wires to refresh the list reflecting task state caused by user interaction in Task Information widget. Specifically, add explicit wires matching

the previous wiring for the incoming events of the deprecated widget to the com.ibm.widget.Refresh event of the Tasks List widget:
- `com.ibm.task.TaskCreated`
- `com.ibm.task.TaskActivated`
- `com.ibm.task.TaskClaimed`
- `com.ibm.task.TaskReleased`
- `com.ibm.task.TaskCompleted`
- `com.ibm.task.TaskDelegated`
- `com.ibm.task.TaskTerminated`
- `com.ibm.task.TaskDeleted`

9. Add explicit wires to highlight the task in the list that is focused in the Task Information or Human Workflow Diagram widget
   - From the `com.ibm.widget.TabChanged` event of the Task Information widget, add a wire to the `com.ibm.widget.Highlight` event of the Tasks List widget.
   - From the `com.ibm.widget.FocusChanged` event of the Human Workflow Diagram widget, add a wire to the `com.ibm.widget.Highlight` event of the Tasks List widget.

Instead of using the Create Tasks widget use the new Task Definitions List widget.

1. Check and capture your specific configuration options for this widget.
2. If you have configured business category filters, define and deploy a query table with the corresponding filter.
3. Check and capture in the Widget Wiring Editor any explicit wiring from and to the deprecated widget.
4. Delete the widget from the page.
5. Add the Task Definitions List widget at the same position onto the page.
6. Configure the widget to match the configuration of the deprecated widget.
7. If you have configured business category filters, configure the task lists for the corresponding query tables.
8. Configure the scenario context to create tasks, services, and processes

Interface maps

The interface map component has been deprecated.

Recommended migration action:

You can migrate your existing interface map modules in WebSphere Integration Developer to use the functions in the mediation flow component.

Service Data Objects

The following Service Data Objects method is deprecated:
- `com.ibm.websphere.sca.sdo.DataFactory.create(Class interfaceClass);`

Recommended migration action:

This method raises "function not supported" exceptions if it is called using the business object framework version 7.0. It will continue to work when invoked using the business object framework version 6.2.

Business Flow Manger

These EJB methods are deprecated, and the corresponding methods you would use, are listed:

Table 3. Business Flow Manager deprecated methods and associated methods which to migrate

Deprecated method	Recommended method for migration
`interface com.ibm.bpe.api. ExpirationBehavior`	`interface com.ibm.bpe.api. TimerBehavior`
`enum RESCHEDULE in com.ibm.bpe. api.ActivityInstanceActions`	`enum RESCHEDULE_TIMER in com. ibm.bpe.api.ActivityInstanceActions`
`enum RESCHEDULE in com.ibm.bpe. api.ActivityInstanceActionIndex`	`enum RESCHEDULE_TIMER in com. ibm.bpe.api.ActivityInstanceActionIndex`
`Enum REASON_POTENTIAL_SENDER in com.ibm.bpe.api.WorkItemData`	No replacement; the method has not yet been used nor are there any plans for it to be used in the future.

The custom property InlineHumanTasks.KeepOverMultipleWhileLoopIterations is deprecated

Recommended migration action:

Use CEI events or audit logging to capture the same information.

This custom property was introduced with version 7.0 to maintain compatibility with previous versions. It affects how Business Process Choreographer handles inline human tasks within loops. The behavior before version 7.0 is incorrect, however some users might rely on this behavior. When this property is not set, inline human tasks within loops cannot be used to get historical information.

HTTPdatabinding

The deprecated HTTPdatabinding methods, and their recommended migration methods, are listed:

Table 4. HTTPdatabinding deprecated methods and associated methods which to migrate

Deprecated method	Recommended method for migration
HTTP SOAP message data binding `com.ibm.websphere.http.data. bindings.HTTPStreamDataBinding SOAP`	`SOAPDataHandler`
HTTP XML message data `bindingcom.ibm.websphere.http. data.bindings.HTTPStreamData BindingXML`	`UTF8XMLDataHandler`
HTTP service gateway message data binding `com.ibm.websphere.http.data. bindings.HTTPServiceGateway DataBinding`	You can use a single data handler that processes all incoming messages across Web Services, HTTP, JMS, and WebSphere MQ called `NativeBodyDataHandler` that function the same as the existing protocol dependant data bindings.

Installation

The IBM Installation Manager is now used to install WebSphere Process Server. It does not have an option to create a deployment environment when you install the product.

Recommended migration action:

You can use the administrative console to configure deployment environments after the product has been installed.

Oracle database support

Oracle version 9 is not supported in version 7.0.

Recommended migration action:

1. If you are using Oracle 9, and have not yet upgraded your database to 10 or 11, perform the upgrade now, as described in the Oracle documentation.
2. If you are using the ojdbc14.jar or the ojdbc5.jar driver, you must install the new ojdbc6.jar driver in the directory that is pointed to by the `ORACLE_JDBC_DRIVER_PATH` WebSphere variable.

Data direct driver bundled with WebSphere Application Server

The embedded data direct driver bundled with WebSphere Application Server is not supported with WebSphere Process Server version 7.0. You must either buy a license for the embedded data direct driver or download the Microsoft JDBC driver for MSSQL Server, which is available for free.

Business Process Choreographer administrative scripts

The tables list the deprecated ProcessContainer MBean method and its administrative script parameters, along with their recommended replacements.

Table 5. ProcessContainer MBean method

Deprecated method	**Recommended method for migration**
`ProcessContainer MBean method deleteCompletedProcessInstances (String state, templateName, validFrom, completedBefore, startedBy)`	`ProcessContainer MBean method deleteCompletedProcessInstances (String[] states, templateName, validFrom, completedAfter, completedBefore, startedBy)`

Table 6. Script parameters

Deprecated parameters	**Replacement parameters**
`deleteAuditLog.py` script parameters **-time** and **processtime**.	Use **-timeUTC** and **-processtimeUTC**
`deleteCompletedProcessInstances.py` script parameters **-validFrom** and **-completedBefore**	Use these parameters: **-validFromUTC** and **-completedBeforeUTC**.
`deleteInvalidProcessTemplate.py` script parameter **-validFrom**	Use **-validFromUTC**.

Table 6. Script parameters (continued)

Deprecated parameters	Replacement parameters
`deleteInvalidTaskTemplate.py` script parameter **-validFrom**	Use **-validFromUTC**
`observerDeleteProcessInstance Data.py` script parameters **-validFrom**, **-deletedBefore**, and **-reachedBefore**	**-validFromUTC**, **-deletedBeforeUTC**, and **-reachedBeforeUTC**

Human Task Manager

The table lists the methods deprecated for the Human Task Manager, and the replacement methods that you use when you migrate your modules.

Table 7. Human Task Manager deprecated methods and associated methods which to migrate

Deprecated method	Recommended method for migration
`HumanTaskManager.getAbsence()`	`HumanTaskManager.getUserSubstitution Detail()`
`HumanTaskManager.getAbsence (String userID)`	`HumanTaskManager.getUserSubstitution Detail(String userID)`
`HumanTaskManager.getSubstitutes()`	`HumanTaskManager.getUserSubstitution Detail()`
`HumanTaskManager.getSubstitutes (String userID)`	`HumanTaskManager.getUserSubstitution Detail(String userID)`

`HumanTaskManager.setAbsence (boolean absence)`	`Sequence:`
	`UserSubstitutionDetail retrievedDetail = HumanTaskManager.getUserSubstitution Detail();retrievedDetail.setStartDate( ..);retrievedDetail.setEndDate(..); HumanTaskManager.setUserSubstitution Detail(retrievedDetail);`

`HumanTaskManager.setAbsence( String userID, boolean absence)`	`Sequence:`
	`UserSubstitutionDetail retrievedDetail = HumanTaskManager.getUserSubstitution Detail(userID);retrievedDetail.setStart Date(..);retrievedDetail.setEndDate(..); HumanTaskManager.setUserSubstitution Detail(userID, retrievedDetail);`

`HumanTaskManager.setSubstitutes (List substitutes)`	`Sequence:`
	`UserSubstitutionDetail retrievedDetail = HumanTaskManager.getUserSubstitution Detail();retrievedDetail.setSubstitutes (..);HumanTaskManager.setUserSubstitution Detail(retrievedDetail);`

`HumanTaskManager.setSubstitutes( String userID, List substitutes)`	`Sequence:`
	`UserSubstitutionDetail retrievedDetail = HumanTaskManager.getUserSubstitution Detail(userID);retrievedDetail.set Substitutes(..);HumanTaskManager.set UserSubstitutionDetail(userID, retrieved Detail);`

Deprecated features in WebSphere Process Server version 6.2

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 stand-alone 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 'cdbSchemaName'):

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

- `public String observerRemoveDeletedInstancesData(String dataSourceName, String cdbSchemaName, String completedBefore)`
- `public String observerRemoveInstanceDataOfTemplate(String dataSourceName, String cdbSchemaName, 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 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)

SCA Admin Commands

The following commands (used with wsadmin) are deprecated:

- configSCAForServer
- configSCAForCluster

Recommended migration action:

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

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

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

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

The following JACL scripts are deprecated:

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

Recommended migration action:

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

Deprecated features in WebSphere Process Server version 6.0.1

WebSphere Process Server version 6.0.1 has no deprecated features.

Deprecated features in WebSphere Process Server version 6.0

Application programming model and container support features

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

Recommended migration action:

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

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

Recommended migration action:

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

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

Recommended migration action:

Use BPEL compensation for business processes.

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

- The WorkList object has been renamed to StoredQuery; consequently, the following methods are deprecated on the BusinessFlowManager bean, and, if applicable, the methods you would use WebSphere Process Server version 6.0 are given:
  - newWorkList(String workListName, String selectClause, String whereClause, String orderByClause, Integer threshold, TimeZone timezone)

    Replace with: createStoredQuery(String storedQueryName, String selectClause, String whereClause, String orderByClause, Integer threshold, TimeZone timezone)
  - getWorkListNames()

Replace with: getStoredQueryNames()

  - deleteWorkList(String workListName)

    Replace with: deleteStoredQuery(String storedQueryName)
  - getWorkList(String workListName)

    Replace with: getStoredQuery(String storedQueryName)
  - executeWorkList(String workListName)

    Replace with: query(String storedQueryName, Integer skipTuples)
  - getWorkListActions()

    not supported.
- The WorkListData object is deprecated.

  Use StoredQueryData instead.
- The following methods of the ProcessTemplateData object are no longer supported:

  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.

Migrating: Heritage 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 Business Integration Server Foundation versions 5.1 and 5.1.1. For more information see “Migrating from WebSphere Studio Application Developer Integration Edition.”
- WebSphere MQ Workflow version 3.6. For more information, see “Migrating from WebSphere MQ Workflow” on page 92.

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 “Migration overview” on page 1 in the Migration: Version-to-version section of the WebSphere Process Server information center.

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

WebSphere Integration Developer contains migration tools that assist in migrating existing application source artifacts 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 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

Printed in USA