IBM WebSphere Process Server for Multiplatforms

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 Multiplatforms (product number 5724-L01) and to all subsequent releases and modifications until otherwise indicated in new editions.

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

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

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 131.

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.

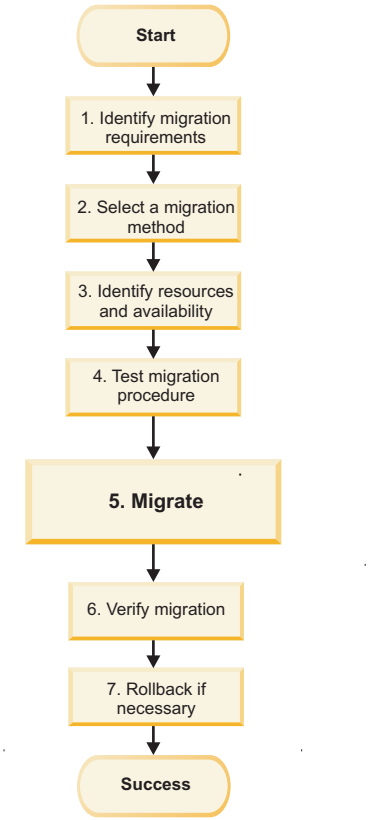

Figure 1. WebSphere Process Server migration roadmap for version-to-version migration

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 15
- “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 BPM profile migration wizard that can be invoked via the BPMMigrate command-line utility or the following set of profile migration command-line tools:

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

Note: If you are migrating from WebSphere Enterprise Service Bus version 6.0.2, you must use the Migrating a profile using the BPM migration command-line utilities procedure.

Note: The BPM profile migration wizard is supported on the following platforms:

- Windows x86 (32-bit)
- Windows x64 (64-bit)
- Linux x86 (32-bit)
- Linux x86-64 (64-bit)
- Linux PPC (32-bit only)
- AIX PPC (32-bit only)
- Solaris SPARC (32-bit only)
- HP-Unix IA64 (64-bit)

Other platforms must use the command-line tools instead of the BPM profile migration wizard to perform the three step profile migration process.

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 102 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 102 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 102 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 BPM Profile Migration Wizard or 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 BPM Profile Migration Wizard or 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. To make sample applications available for a stand-alone profile, you can install them using the installation wizard for the later version of WebSphere Process Server. For network deployment profiles, any samples installed with the previous version of WebSphere Process Server will be installed during migration to the new version.

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 (wizard or 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 93.

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 93.

  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

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 Prerequisites for installing 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 or a configurable value in the BPM profile migration wizard 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 or the BPM profile migration wizard 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, a migration to a remote system, or a migration that

requires the operating system on the source system to be upgraded during the migration process. 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.

About runtime migration procedures

The runtime migration tools and documentation support the following three migration procedures: migrating stand-alone environments, migrating network deployment environments with full downtime, and migrating network deployment environments with minimal downtime.

Each of the three runtime migration procedures contains a set of steps and subprocedures. In addition to understanding how the procedures work, it is equally important to consider how you will test the migration procedure you select. The following sections provide an overview of each procedure as well as information to consider about testing the migration.

- "Migrating stand-alone environments"
- "Migrating network deployment environments with full downtime" on page 31
- "Migrating network deployment environments with minimal downtime" on page 31
- "Migration testing" on page 31

Migrating stand-alone environments

The procedure for migrating stand-alone environments describes the steps for backing up the environment, migrating the stand-alone profile, and upgrading the product databases configured for the profile. The procedure contains variants for the different supported mechanisms of migrating a stand-alone environment including side-by-side migration, remote migration, and operating system upgrade migration. Before migrating a stand-alone environment, determine which of these variants fits your requirements best.

For instructions on using this procedure, see “Migrating a stand-alone environment” on page 32.

Migrating network deployment environments with full downtime

There are two different procedures for migrating network deployment environments that differ depending on the length of your downtime migration window. The full downtime procedure is the simplest procedure and is recommended if your downtime window can accommodate the migration. The length of the migration will depend on several factors including the source version, the number of cells, clusters, nodes, applications, and the amount of data in the database. To determine the length your migration will take use the full migration process in your staging environment. It is critical that you follow the network deployment procedure steps carefully and in the order they are listed to ensure that you successfully migrate your network deployment environment.

For instructions on using this procedure, see “Migrating a network deployment environment with full downtime” on page 37.

Migrating network deployment environments with minimal downtime

The minimal downtime procedure should be used if you are unable to accommodate the migration using the full downtime procedure for your migration window and you can accommodate the downtime required for the minimal downtime procedure or in scenarios where the amount of downtime required for the migration directly impacts your business. The minimal downtime procedure is more complex than the full downtime procedure and should only be used when the length of the downtime is critical. If you are not able to accommodate minimal downtime you should consider either the manual or artifact migration methods instead of the runtime migration method. The minimal downtime procedure involves splitting the migration into two groups and migrating one group while the other is still running minimizing the downtime for the cluster. The minimal downtime occurs just prior to bringing the migrated group of nodes online in order to update the database schema and the product data.

Note: If the source version contains applications that exploit Business Calendars or mediation flow components, the minimum downtime procedure cannot be used unless those applications can tolerate some downtime. Nodes with servers that are running applications that exploit Business Calendars or mediation flow components will remain stopped until the node is migrated to version 7.0.

For instructions on using this procedure, see “Migrating a network deployment environment with minimal downtime” on page 48.

Migration testing

It is critical that any production migrations are thoroughly tested in a staging environment before they are attempted in a production setting. In addition, it is important that the backup steps in the procedures are followed carefully to enable rollback in cases where the configuration data or the applications failed to migrate successful to the target environment. Manual and artifact migration methods are often used in conjunction with runtime migration to validate that a typical application or all the applications can be deployed to a version 7.0 environment without issue or the applications can be migrated by the development tools successfully thus providing greater assurance that backwards compatibility for the

application will be maintained. If you intend to migrate a network deployment environment, it is also helpful to start with a stand-alone environment in a staging environment to learn how to use the tools and the essence of the runtime migration process before using the more involved network deployment full downtime or minimal downtime procedures.

Migrating a stand-alone environment

Use this procedure to migrate a stand-alone environment.

Before you begin

Review the "Migration overview" on page 1 and BPM runtime premigration checklist topics.

About this task

To migrate a stand-alone environment, use the following procedure.

Procedure

1. **Install the migration target product(s).**
   - For a side-by-side migration, install the target product and latest fix packs on the same system as the source product of the migration.
   - For a remote migration, install the target product and latest fix packs on the system that will serve as the target for the migration.
   - For an operating system upgrade migration, defer the installation until after the operating system is upgraded.

   Note: You must either install the target version with the same user ID as that used for installing the source version or have permission to access the configuration and data on the source installation.

   Note: To migrate from source profiles augmented by multiple products, the new version of those products must be installed into the same target installation directory. For example, if the source profile is augmented by WebSphere Process Server and WebSphere Business Monitor, both of those products must be installed into the same target installation directory.
2. **Upgrade 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, perform the upgrade now, as described in the DB2 for z/OS documentation.
3. **Upgrade Oracle 9i and the Oracle JDBC driver.**

   Important: You must perform this step on all WebSphere Process Server installations that access the Oracle database.

   a. If you are using Oracle 9i and have not yet upgraded your database to 10g or 11g, perform the upgrade now, as described in the Oracle documentation.

   b. 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. To do this, use the following procedure.

      1) Check the value for the *ORACLE_JDBC_DRIVER_PATH* variable in the previous environment. Use one of the following methods to do this:

- In the administrative console, select **Environment → WebSphere variables**, then select the scope that matches the node of the source profile.
- Navigate to the `variables.xml` file in the following directory: *source_profile_root*`\config\cells\`*cell_name*`\nodes\`*node_name*`\`.

Note: The cell name and node name must match the source profile information.

2) Install the new `ojdbc6.jar` driver in the directory that is pointed to by the *ORACLE_JDBC_DRIVER_PATH* WebSphere variable. Use one of the following steps, depending on the location pointed to by the variable.
   - If the variable points to a directory outside of the WebSphere Process Server installation, copy the `odbc6.jar` file into the same folder where the `ojdbc14.jar` or `ojdbc5.jar` file resides.
   - If the variable points to a directory within the WebSphere Process Server installation, create the same directory structure in the WebSphere Process Server version 7.0 installation, and then copy the `odbc6.jar` file into that directory.

4. **Stop the migration source server.**

   Stop the migration source server using the `stopServer` command from the *profile_root*`/bin` directory on the migration source system or from the profile's First steps console. Use the following syntax:

   - Linux UNIX **On Linux® and UNIX® platforms:** `stopServer.sh` *server_name* `-username` *user_name* `-password` *password*
   - Windows **On Windows® platforms:** `stopServer.bat` *server_name* `-username` *user_name* `-password` *password*

   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.

5. **Backup the migration source profile.**

   Back up the profile configuration on the migration source server using the `backupConfig` command.

   Use the following syntax to backup a profile named profile1 to /ProfileBackups/profile1.zip.

   - Linux UNIX **On Linux and UNIX platforms:** `backupConfig.sh /ProfileBackups/profile1.zip -profileName profile1`
   - Windows **On Windows platforms:** `backupConfig.bat c:\ProfileBackups\profile1.zip -profileName profile1`

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

6. **Back up the .nifRegistry file.**

   The .nifRegistry file identifies the installation root for all installed WebSphere Process Server products; it also identifies the installation root for all installed WebSphere Application Server products. It is located as follows:

   - Linux UNIX **On Linux or UNIX platforms**: `/opt/.ibm/.nif/.nifregistry`
   - Windows **On Windows platforms**:
     - If the user ID that installed the product had administrative privileges, the file is located in the Windows root directory (`C:\Windows` or `C:\WINNT` on most Windows systems).
     - If the user ID that installed the product did not have administrative privileges, the file is located in the home directory of that user ID.

7. **Back up the migration source product databases.**

   Back up the following databases that are configured by the stand-alone profile according to the documentation for your database:

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

8. **Migrate the stand-alone server profile.**
   - For a side-by-side migration, the BPM profile migration wizard or the BPM migration command-line utilities can be used to migrate the source profile. However, if you are migrating from WebSphere Enterprise Service Bus version 6.0.2, you must use the Migrating a profile using the BPM migration command-line utilities procedure.
     - To use the BPM profile migration wizard, follow the "Migrating a profile using the BPM profile migration wizard" on page 63 procedure on the system containing the source profile.
     - To use the BPM migration command line utilities, follow the Migrating a profile using the BPM migration command-line utilities procedure on the system containing the source profile.
   - For a remote migration, follow the Migrating a stand-alone profile to a remote system procedure. If the source operating system is i5/OS, the remote migration procedure is necessary to migrate the stand-alone profile to a supported operating system.
   - For an operating system upgrade migration, see the Migrating a stand-alone profile while upgrading an operating system procedure.

9. **Update Cloudscape or Derby databases.**

   If you are using Cloudscape or Derby databases, you must ensure that you are using the supported version. For instructions on migrating Cloudscape to Derby, see Migrating IBM Cloudscape or Apache Derby databases

10. **Copy the database migration and upgrade scripts to your database system.**

    On the target migration system, find the database migration and upgrade commands and scripts that are customized to your database type and copy them to your database system. The commands and scripts are located in the following directories:

- **Common database**: *install_root*/dbscripts/CommonDB/*database_type*
- **Business Space database**:
  - **Stand-alone server**: *profile_root*/dbscripts/BusinessSpace/*node_name_server_name/database_product_name/database_name*
  - **Cluster**: *profile_root*/dbscripts/BusinessSpace/*cluster_name/database_product_name/database_name*

Note: The Business Process Choreographer database commands and scripts are copied in a later process using the DBDesignGenerator command. For more information, see "Upgrading the Business Process Choreographer database schema" on page 75.
Use the following table to determine the directory name that corresponds to your particular database type:

Database type	Directory name
DB2 Universal Database™ (for all operating systems except z/OS® and i5/OS®)	DB2
DB2 Universal Database for i5/OS	DB2iSeries
DB2® for z/OS Version 8.x	DB2zOSV8 - Use scripts in this directory if your initial database configuration used DB2 z/OS v8 (uses long table names) or if you upgraded from DB2 z/OS v7 to DB2 z/OS v8
DB2 for z/OS Version 9.x	DB2zOSV9 - Use scripts in this directory if your initial database configuration used DB2 z/OS v9 or later (uses long table names) or if you upgraded from DB2 z/OS v7 to DB2 z/OS v9.
Derby	Derby 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.
Informix®	Informix
Oracle	Oracle
Microsoft® SQL Server	SQLServer

11. **Update the data source configuration.** If you have data sources using the embedded data direct driver and you did not update them in the source environment to use a licensed Data Direct JDBC driver or Microsoft JDBC driver, update the Data Source configuration. To do this use the following procedure.

    Attention: The SystemOut.log file might reflect errors because some components could not establish connection to the database.

    a. **Start the migration 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. Use the following syntax:

- Linux UNIX **On Linux and UNIX platforms:** `startServer.sh server_name`
- Windows **On Windows platforms:** `startServer.bat server_name`

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

b. Log in to the administrative console.

c. Update the data source configuration using 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 embedded driver.

3) Use the Test Connection option to check if the data source configuration works.

4) Stop the migration source server.

Stop the migration source server using the `stopServer` command from the *profile_root*`/bin` directory on the migration source system or from the profile's First steps console. Use the following syntax:

- Linux UNIX **On Linux and UNIX platforms:** `stopServer.sh server_name -username user_name -password password`
- Windows **On Windows platforms:** `stopServer.bat server_name -username user_name -password password`

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.

12. **Migrate the product databases.**

Migrate each of the product databases that are configured on the stand-alone server using the following procedures:

a. Upgrade the Common database schema manually using the Upgrading the Common database schema procedure if the database user that is defined for the Common database data source does not have sufficient privileges.

b. Upgrade the Business Process Choreographer Database schema manually using the Upgrading the Business Process Choreographer database schema procedure.

c. Migrate the Business Process Choreographer database data using the Migrating the Business Process Choreographer database data procedure if your source version you are migrating from is 6.0.2, 6.1.0, or 6.1.2.

d. Migrate the Business Space database schema using the Migrating the Business Space database schema procedure.

e. Migrate the Business Space database data using the Migrating the Business Space Database data procedure.

f. Optional: Migrate the messaging engine database if it is needed for your environment. To learn more about when and how to migrate the messaging engine, see Migrating a messaging engine based on a data store on the WebSphere Application Server Version 7.0 information center.

13. **Start the migration 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. Use the following syntax:

- Linux UNIX **On Linux and UNIX platforms:** `startServer.sh` *server_name*
- Windows **On Windows platforms:** `startServer.bat` *server_name*

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

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 86.

Migrating a network deployment environment with full downtime

Use this procedure to migrate a network deployment environment while incurring full downtime.

Before you begin

Review the "Migration overview" on page 1 and BPM runtime premigration checklist topics.

About this task

Follow these steps to migrate a network deployment environment while incurring full downtime.

Procedure

1. **Install the migration target product(s).**

Install the target product and latest fix packs on the same system as the source product of the migration.

Note: You must either install the target version with the same user ID as that used for installing the source version or have permission to access the configuration and data on the source installation.

Note: To migrate from source profiles augmented by multiple products, the new version of those products must be installed into the same target installation directory. For example, if the source profile is augmented by WebSphere Process Server and WebSphere Business Monitor, both of those products must be installed into the same target installation directory.

2. **Upgrade 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, perform the upgrade now, as described in the DB2 for z/OS documentation.

3. **Upgrade Oracle 9i and the Oracle JDBC driver.**

   Important: You must perform this step on all WebSphere Process Server installations that access the Oracle database.

   a. If you are using Oracle 9i and have not yet upgraded your database to 10g or 11g, perform the upgrade now, as described in the Oracle documentation.

   b. 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. To do this, use the following procedure.

      1) Check the value for the *ORACLE_JDBC_DRIVER_PATH* variable in the previous environment. Use one of the following methods to do this:
         - In the administrative console, select **Environment → WebSphere variables**, then select the scope that matches the node of the source profile.
         - Navigate to the `variables.xml` file in the following directory: *source_profile_root*`\config\cells\`*cell_name*`\nodes\`*node_name*`\`.

         Note: The cell name and node name must match the source profile information.

      2) Install the new `ojdbc6.jar` driver in the directory that is pointed to by the *ORACLE_JDBC_DRIVER_PATH* WebSphere variable. Use one of the following steps, depending on the location pointed to by the variable.
         - If the variable points to a directory outside of the WebSphere Process Server installation, copy the `odbc6.jar` file into the same folder where the `ojdbc14.jar` or `ojdbc5.jar` file resides.
         - If the variable points to a directory within the WebSphere Process Server installation, create the same directory structure in the WebSphere Process Server version 7.0 installation, and then copy the `odbc6.jar` file into that directory.

4. **Identify the clusters, clustered managed nodes, and non-clustered managed nodes to migrate.**

   If you intend to migrate an entire cell, you will migrate:
   - the deployment manager,
   - all nodes that do not have an application server that is a member of any cluster in the cell (non-clustered managed nodes),
   - all clusters and all the nodes that have application servers that are members of those clusters (clustered managed nodes).

   If you are **not** migrating an entire cell, you **do not** intend to migrate any clusters, and you **do** intend to migrate one or more nodes that do not have an application server that is a member of any cluster in the cell (non-clustered managed nodes), you will migrate:

- the deployment manager,
- and each non-clustered managed node you intend to migrate.

If you are **not** migrating an entire cell, intend to migrate one or more clusters in the cell, and zero or more non-clustered managed nodes, you will migrate:

- the deployment manager,
- each non-clustered managed node you intend to migrate,
- each cluster you explicitly intend to migrate and all the nodes that have an application server that is a member of that cluster (clustered managed nodes),
- and any cluster and that cluster's clustered managed nodes implicitly impacted by the clusters that you intend to migrate. To identify the transitive closure of all the impacted clusters and their clustered managed nodes, use the following procedure:
  - For each cluster you intend to migrate, identify all the clustered managed nodes that have application servers that contribute to the cluster.
  - For each clustered managed node, determine what other clusters, if any, the application servers running on the node are members of.
  - Repeat the process for each of these clusters to determine the complete set of clusters and clustered managed nodes that must be migrated as part of this procedure.

5. **Stop the deployment manager.**

   Stop the source deployment manager using the `stopManager` command from the *`profile_root`*`/bin` directory on the migration source system or from the profile's First steps console.

   Use the following syntax:

   - Linux UNIX **On Linux and UNIX platforms:** `stopManager.sh -username` *`user_name`* `-password` *`password`*
   - Windows **On Windows platforms:** `stopManager.bat -username` *`user_name`* `-password` *`password`*

   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 `stopManager` command, see the stopManager command topic on the WebSphere Application Server, Version 7.0 information center.

6. **Stop the non-clustered managed node migration source servers.**

   Repeat this step for each server associated with a non-clustered managed node that will be migrated.

   Stop the migration source server using the `stopServer` command from the *`profile_root`*`/bin` directory on the migration source system or from the profile's First steps console.

   Use the following syntax:

- Linux UNIX **On Linux and UNIX platforms:** `stopServer.sh server_name -username user_name -password password`
- Windows **On Windows platforms:** `stopServer.bat server_name -username user_name -password password`

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.

7. **Stop the non-clustered managed node migration source node agents.**

   Repeat this step for each node agent associated with a non-clustered managed node that will be migrated.

   Stop the migration source's node agent using the `stopNode` command from the *profile_root*`/bin` directory of the migration source system.

   Use the following syntax:

   - Linux UNIX **On Linux and UNIX platforms:** `stopNode.sh -username user_name -password password`
   - Windows **On Windows platforms:** `stopNode.bat -username user_name -password password`

   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 system.

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

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

8. **Stop the clustered managed node migration source servers.**

   Repeat this step for each server associated with a clustered managed node that will be migrated.

   Stop the migration source server using the `stopServer` command from the *profile_root*`/bin` directory on the migration source system or from the profile's First steps console.

   Use the following syntax:

   - Linux UNIX **On Linux and UNIX platforms:** `stopServer.sh server_name -username user_name -password password`
   - Windows **On Windows platforms:** `stopServer.bat server_name -username user_name -password password`

   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.

9. **Stop the clustered managed node migration source node agents.**

   Repeat this step for each node agent associated with a clustered managed node that will be migrated.

   Repeat this step for each node agent that is impacted by the migration.

   Stop the migration source's node agent using the `stopNode` command from the *profile_root*`/bin` directory of the migration source system.

   Use the following syntax:

   - Linux UNIX **On Linux and UNIX platforms:** `stopNode.sh -username` *user_name* `-password` *password*
   - Windows **On Windows platforms:** `stopNode.bat -username` *user_name* `-password` *password*

   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 system.

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

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

10. **Backup the migration source profiles.**

    Repeat this step for each profile that will be migrated, including the deployment manager, each non-clustered managed node, and each managed node.

    Back up the profile configuration on the migration source server using the `backupConfig` command.

    Use the following syntax to backup a profile named profile1 to /ProfileBackups/profile1.zip.

    - Linux UNIX **On Linux and UNIX platforms:** `backupConfig.sh /ProfileBackups/profile1.zip -profileName profile1`
    - Windows **On Windows platforms:** `backupConfig.bat c:\ProfileBackups\profile1.zip -profileName profile1`

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

11. **Back up the .nifRegistry file.**

The .nifRegistry file identifies the installation root for all installed WebSphere Process Server products; it also identifies the installation root for all installed WebSphere Application Server products. It is located as follows:

- Linux UNIX **On Linux or UNIX platforms**: `/opt/.ibm/.nif/.nifregistry`
- Windows **On Windows platforms**:
  - If the user ID that installed the product had administrative privileges, the file is located in the Windows root directory (`C:\Windows` or `C:\WINNT` on most Windows systems).
  - If the user ID that installed the product did not have administrative privileges, the file is located in the home directory of that user ID.

12. **Backup the migration source product databases.**

    Backup the following databases that are configured by any of the migration source profiles according to the documentation for your database:

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

13. **Migrate the deployment manager profile.**

    The BPM profile migration wizard or the BPM migration command-line utilities can be used to migrate the deployment manager source profile. To use the BPM profile migration wizard, follow the "Migrating a profile using the BPM profile migration wizard" on page 63 subprocedure on the system containing the deployment manager profile. To use the BPM migration command-line utilities, follow the Migrating a profile using the BPM migration command-line utilities subprocedure on the system containing the deployment manager profile.

    Note: If you are migrating from WebSphere Enterprise Service Bus version 6.0.2, you must use the "Migrating a profile using the BPM profile migration wizard" on page 63 procedure.

14. **Update Cloudscape or Derby databases.**

    If you are using Cloudscape or Derby databases, you must ensure that you are using the supported version. For instructions on migrating Cloudscape to Derby, see Migrating IBM Cloudscape or Apache Derby databases

15. **Copy the Common database migration and upgrade scripts to your database system.**

    On the target migration system, find the Common database migration and upgrade commands and scripts that are customized to your database type and copy them to your database system. The commands and scripts are located in the following directories: *install_root*`/dbscripts/CommonDB/`*database_type*

    Note: The Business Process Choreographer database commands and scripts are copied in a later process using the DBDesignGenerator command. For more information, see "Upgrading the Business Process Choreographer database schema" on page 75.
    Use the following table to determine the directory name that corresponds to your particular database type:

Database type	Directory name
DB2 Universal Database (for all operating systems except z/OS and i5/OS)	`DB2`
DB2 Universal Database for i5/OS	`DB2iSeries`
DB2 for z/OS Version 8.x	`DB2zOSV8` - Use scripts in this directory if your initial database configuration used DB2 z/OS v8 (uses long table names) or if you upgraded from DB2 z/OS v7 to DB2 z/OS v8
DB2 for z/OS Version 9.x	`DB2zOSV9` - Use scripts in this directory if your initial database configuration used DB2 z/OS v9 or later (uses long table names) or if you upgraded from DB2 z/OS v7 to DB2 z/OS v9.
Derby	`Derby` 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.
Informix	`Informix`
Oracle	`Oracle`
Microsoft SQL Server	`SQLServer`

16. **Upgrade the cell scoped Common database.**

    Upgrade the Common database schema manually using the Upgrading the Common database schema procedure if the database user that is defined for the Common database data source does not have sufficient privileges.

17. **Start the target deployment manager.**

    Start the target deployment manager using the `startManager` command from the *`profile_root`*`/bin` directory on the deployment manager system or from the deployment manager profile's First steps console.

    Use the following syntax:

    - Linux UNIX **On Linux and UNIX platforms:** `startManager.sh`
    - Windows **On Windows platforms:** `startManager.bat`

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

18. **Update the data source configuration.** If you have data sources using embedded data direct driver and you did not update them in the source environment to use a licensed Data Direct JDBC driver or Microsoft JDBC driver, update the Data Source configuration. To do this, use the following procedure.

    Attention: The `SystemOut.log` file might reflect errors because some components could not establish connection to the database.

    a. Log in to the administrative console.

b. 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.

c. Delete the existing Data Source that uses embedded driver.

d. Use the Test Connection option to check if the data source configuration works.

e. Restart the deployment manager.

19. **Migrate the non-clustered managed nodes.**

This step should be repeated for each non-clustered managed node that is intended to be migrated.

The BPM profile migration wizard or the BPM migration command-line utilities can be used to migrate the non-clustered managed node source profile. To use the BPM profile migration wizard, follow the "Migrating a profile using the BPM profile migration wizard" on page 63 subprocedure on the system containing the non-clustered managed node profile. To use the BPM migration command-line utilities, follow the Migrating a profile using the BPM migration command-line utilities subprocedure on the system containing the non-clustered managed node profile.

Note: If you are migrating from WebSphere Enterprise Service Bus version 6.0.2, you must use the Migrating a profile using the BPM migration command-line utilities procedure.

20. **Migrate the non-clustered managed node product databases.**

This step should be repeated for each non-clustered managed node that is intended to be migrated.

Migrate each of the product databases that are configured on the non-clustered managed node using the following procedures:

a. Upgrade the Business Process Choreographer Database schema manually using the Upgrading the Business Process Choreographer database schema procedure if either of the following conditions are true:

- You did not use the default table spaces for the Business Process Choreographer database. If you used the sample Business Process Choreographer configuration, or have created all of the database objects in the default table spaces specified in the sample SQL scripts, your database uses the default table spaces. This is typically the case for a test environment.
- The database user that is configured for the BPEDB data source is not authorized to perform all of the following operations: create and alter tables, create and drop indexes and views, and for the table SCHEMA_VERSION: query, update, delete, and insert.

b. Migrate the Business Process Choreographer database data using the Migrating the Business Process Choreographer database data procedure if your source version you are migrating from is 6.0.2, 6.1.0, or 6.1.2.

c. Migrate the Business Space database schema using the Migrating the Business Space database schema procedure.

d. Migrate the Business Space database data using the Migrating the Business Space Database data procedure.

e. Optional: Migrate the messaging engine database if it is needed for your environment. To learn more about when and how to migrate the

messaging engine, see Migrating a messaging engine based on a data store on the WebSphere Application Server Version 7.0 information center.

21. **Optional: Migrate the Business Rules Managers.**

    This step should be repeated for each non-clustered managed node that is intended to be migrated.

    The Business Rules Manager is automatically migrated when the last node in the cell is migrated but if the migrated non-clustered managed node contains the Business Rules Manager it can manually be migrated.

    To manually migrate the Business Rules Manager for the server server1 and the non-clustered managed node node1 use the following command:

    ```
    wsadmin -f installBRManager.jacl -s server1 -n node1
    ```

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

22. **Migrate the clusters.**

    Repeat the following procedure for each cluster in the network deployment environment that needs to be migrated.

    a. **Migrate the managed nodes.**

       This step should be repeated for each managed node in the cluster.

       The BPM profile migration wizard or the BPM migration command-line utilities can be used to migrate the managed node source profile. To use the BPM profile migration wizard, follow the "Migrating a profile using the BPM profile migration wizard" on page 63 procedure on the system containing the source profile. To use the BPM migration command-line utilities, follow the Migrating a profile using the BPM migration command-line utilities subprocedure on the system containing the source profile.

       Note: If you are migrating from WebSphere Enterprise Service Bus version 6.0.2, you must use the Migrating a profile using the BPM migration command-line utilities procedure.

    b. **Migrate the cluster scoped profile.**

       Migrate the cluster scoped profile using the `BPMMigrateCluster` command from the *`install_root`*`/bin` directory on the system containing the deployment manager.

       Use the following syntax to migrate a cluster named applicationCluster1 with a deployment manager profile named dmgrProfile copied in the /MigrationSnapshots/ProcServer620 directory:

       - Linux UNIX **On Linux and UNIX platforms:**
         ```
         BPMMigrateCluster.sh /MigrationSnapshots/ProcServer620
         applicationCluster1 dmgrProfile
         ```
       - Windows **On Windows platforms:** `BPMMigrateCluster.bat`
         ```
         c:\MigrationSnapshots\ProcServer620 applicationCluster1
         dmgrProfile
         ```

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

    c. **Back up your managed profiles.**

       Repeat this step for each profile in the managed node. This backup is needed in case the next step for running the syncNode command fails. After resolving the issue with syncNode, you can restore the backup before running syncNode command again.

Back up the profile configuration on the non-clustered managed node using the `backupConfig` command.

Use the following syntax to backup a profile named profile1 to /ProfileBackups/profile1.zip.

- Linux UNIX **On Linux and UNIX platforms:** `backupConfig.sh /ProfileBackups/profile1.zip -profileName profile1`
- Windows **On Windows platforms:** `backupConfig.bat c:\ProfileBackups\profile1.zip -profileName profile1`

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

d. **Synchronize the managed nodes.**

Repeat this step for each managed node in the cluster.

Synchronize the node with the target deployment manager using syncNode command from the *profile_root*`/bin` directory of the migration target profile or from the target profile's First steps console.

Use the following syntax:

- Linux UNIX **On Linux and UNIX platforms:** `syncNode.sh` *deployment_manager_machine_name_or_ip_address deployment_manager_port_no*
- Windows **On Windows platforms:** `syncNode.bat` *deployment_manager_machine_name_or_ip_address deployment_manager_port_no*

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

e. **Migrate the cluster scoped product databases.**

Migrate each of the product databases that are configured for the cluster using the following procedures:

1) Upgrade the Business Process Choreographer Database schema manually using the Upgrading the Business Process Choreographer database schema procedure if either of the following conditions are true:
   - You did not use the default table spaces for the Business Process Choreographer database. If you used the sample Business Process Choreographer configuration, or have created all of the database objects in the default table spaces specified in the sample SQL scripts, your database uses the default table spaces. This is typically the case for a test environment.
   - The database user that is configured for the BPEDB data source is not authorized to perform all of the following operations: create and alter tables, create and drop indexes and views, and for the table SCHEMA_VERSION: query, update, delete, and insert.
2) Migrate the Business Process Choreographer database data using the Migrating the Business Process Choreographer database data procedure if your source version you are migrating from is 6.0.2, 6.1.0, or 6.1.2.
3) Migrate the Business Space database schema using the Migrating the Business Space database schema procedure.
4) Migrate the Business Space database data using the Migrating the Business Space Database data procedure.

5) Optional: Migrate the messaging engine database if it is needed for your environment. To learn more about when and how to migrate the messaging engine, see Migrating a messaging engine based on a data store on the WebSphere Application Server Version 7.0 information center.

f. **Optional: Migrate the Business Rules Manager.**

The Business Rules Manager is automatically migrated when the last node in the cell is migrated but if the migrated cluster contains the Business Rules Manager it can manually be migrated.

To manually migrate the Business Rules Manager for the cluster cluster1 use the following command:

```
wsadmin -f installBRManager.jacl -cl cluster1
```

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

g. **Start the migration target node agents**.

Repeat this step for each non-clustered managed node that was migrated, and each clustered managed node for each cluster that was migrated.

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.

Use the following syntax:

- Linux UNIX **On Linux and UNIX platforms:** `startNode.sh`
- Windows **On Windows platforms:** `startNode.bat`

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

23. **Start the migration target servers**.

Repeat this step for each server configured for each non-clustered managed node that was migrated and for each clustered managed node that was migrated.

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.

Use the following syntax:

- Linux UNIX **On Linux and UNIX platforms:** `startServer.sh` *server_name*
- Windows **On Windows platforms:** `startServer.bat` *server_name*

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

24. **Optional: Uninstall the source deployment manager**.

Once the migration is complete, the migration source deployment manager can be uninstalled.

25. **Remove Compatibility Mode.**

If you chose the compatibility option (which is the default), and if all your nodes are completely migrated to the target version, run the `convertScriptCompatibility` script from the *install_root*`/bin` directory on the deployment manager and each node to remove compatibility.

Use the following syntax:

- Linux UNIX **On Linux and UNIX platforms:** `convertScriptCompatibility.sh`
- Windows **On Windows platforms:** `convertScriptCompatibility.bat`

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

Results

The network deployment environment is migrated to the target version.

What to do next

Verify that the migration was successful. For instructions, see "Verifying migration" on page 86.

Migrating a network deployment environment with minimal downtime

Use this procedure to migrate a network deployment environment while incurring minimal downtime.

Before you begin

Review the "Migration overview" on page 1 and "Runtime premigration checklist" on page 24 topics.

Note: If the source version contains applications that exploit Business Calendars or mediation flow components, the minimum downtime procedure cannot be used unless those applications can tolerate some downtime. Nodes with servers that are running applications that exploit Business Calendars or mediation flow components will remain stopped until the node is migrated to version 7.0.

About this task

Follow these steps to migrate a network deployment environment while incurring minimal downtime.

Procedure

1. **Install the migration target product(s).**

   Install the target product and the latest fix packs on the same system as the source product of the migration.

   Note: You must either install the target version with the same user ID as that used for installing the source version or have permission to access the configuration and data on the source installation.

   Note: To migrate from source profiles augmented by multiple products, the new version of those products must be installed into the same target installation directory. For example, if the source profile is augmented by WebSphere Process Server and WebSphere Business Monitor, both of those products must be installed into the same target installation directory.
2. **Upgrade 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, perform the upgrade now, as described in the DB2 for z/OS documentation.

3. **Upgrade Oracle 9i and the Oracle JDBC driver.**

   Important: You must perform this step on all WebSphere Process Server installations that access the Oracle database.

   a. If you are using Oracle 9i and have not yet upgraded your database to 10g or 11g, perform the upgrade now, as described in the Oracle documentation.

   b. 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. To do this, use the following procedure.

      1) Check the value for the *ORACLE_JDBC_DRIVER_PATH* variable in the previous environment. Use one of the following methods to do this:
         - In the administrative console, select **Environment → WebSphere variables**, then select the scope that matches the node of the source profile.
         - Navigate to the `variables.xml` file in the following directory: *source_profile_root*`\config\cells\`*cell_name*`\nodes\`*node_name*`\`.

         Note: The cell name and node name must match the source profile information.

      2) Install the new `ojdbc6.jar` driver in the directory that is pointed to by the *ORACLE_JDBC_DRIVER_PATH* WebSphere variable. Use one of the following steps, depending on the location pointed to by the variable.
         - If the variable points to a directory outside of the WebSphere Process Server installation, copy the `odbc6.jar` file into the same folder where the `ojdbc14.jar` or `ojdbc5.jar` file resides.
         - If the variable points to a directory within the WebSphere Process Server installation, create the same directory structure in the WebSphere Process Server version 7.0 installation, and then copy the `odbc6.jar` file into that directory.

4. **Identify the clusters, clustered managed nodes, and non-clustered managed nodes to migrate.**

   If you intend to migrate an entire cell, you will migrate:
   - the deployment manager,
   - all nodes that do not have an application server that is a member of any cluster in the cell (non-clustered managed nodes),
   - all clusters and all the nodes that have application servers that are members of those clusters (clustered managed nodes).

   If you are **not** migrating an entire cell, you **do not** intend to migrate any clusters, and you **do** intend to migrate one or more nodes that do not have an application server that is a member of any cluster in the cell (non-clustered managed nodes), you will migrate:
   - the deployment manager,
   - and each non-clustered managed node you intend to migrate.

   If you are **not** migrating an entire cell, intend to migrate one or more clusters in the cell, and zero or more non-clustered managed nodes, you will migrate:
   - the deployment manager,

- each non-clustered managed node you intend to migrate,
- each cluster you explicitly intend to migrate and all the nodes that have an application server that is a member of that cluster (clustered managed nodes),
- and any cluster and that cluster's clustered managed nodes implicitly impacted by the clusters that you intend to migrate. To identify the transitive closure of all the impacted clusters and their clustered managed nodes, use the following procedure:
  - For each cluster you intend to migrate, identify all the clustered managed nodes that have application servers that contribute to the cluster.
  - For each clustered managed node, determine what other clusters, if any, the application servers running on the node are members of.
  - Repeat the process for each of these clusters to determine the complete set of clusters and clustered managed nodes that must be migrated as part of this procedure.

5. **Disable synchronization for all nodes.**

   Disable synchronization for all non-clustered managed nodes and clustered managed nodes using the administration console on the source deployment manager.

   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. Make a note of the following settings so they can be restored later in the procedure when you re-enable node synchronization:
      - **Enable service at server startup**
      - **Automatic synchronization**
      - **Startup synchronization**
   e. De-select the following options:
      - **Enable service at server startup**
      - **Automatic migration**
      - **Startup synchronization**
   f. Click **Apply**, then click **OK** to save the configuration changes and to make sure that all nodes in the cell are synchronized to make the changes effective on the node agents.

6. **Stop the deployment manager.**

   Stop the migration source's deployment manager using the `stopManager` command from the *profile_root*`/bin` directory on the migration source system or from the profile's First steps console.

   Use the following syntax:

   - Linux UNIX **On Linux and UNIX platforms:** `stopManager.sh -username` *user_name* `-password` *password*
   - Windows **On Windows platforms:** `stopManager.bat -username` *user_name* `-password` *password*

   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 `stopManager` command, see the stopManager command topic on the WebSphere Application Server, Version 7.0 information center.

7. **Backup the source deployment manager profile.**

   Back up the deployment manager profile configuration on the source deployment manager system using the `backupConfig` command.

   Use the following syntax to backup a profile named dmgrProfile to /ProfileBackups/dmgrProfile.zip.

   - Linux UNIX **On Linux and UNIX platforms:** `backupConfig.sh /ProfileBackups/dmgrProfile.zip -profileName dmgrProfile`
   - Windows **On Windows platforms:** `backupConfig.bat c:\ProfileBackups\profile1.zip -profileName dmgrProfile`

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

8. **Back up the .nifRegistry file.**

   The .nifRegistry file identifies the installation root for all installed WebSphere Process Server products; it also identifies the installation root for all installed WebSphere Application Server products. It is located as follows:

   - Linux UNIX **On Linux or UNIX platforms**: `/opt/.ibm/.nif/.nifregistry`
   - Windows **On Windows platforms**:
     - If the user ID that installed the product had administrative privileges, the file is located in the Windows root directory (`C:\Windows` or `C:\WINNT` on most Windows systems).
     - If the user ID that installed the product did not have administrative privileges, the file is located in the home directory of that user ID.

9. **Backup the cell scoped Common database.**

   Backup the cell scoped Common database using the documentation for your database server.

10. **Migrate the deployment manager profile.**

    The BPM profile migration wizard or the BPM migration command-line utilities can be used to migrate the deployment manager source profile. To use the BPM profile migration wizard, follow the Migrating a profile using the BPM profile migration wizard subprocedure on the system containing the deployment manager profile. To use the BPM migration command-line utilities, follow the Migrating a profile using the BPM migration command-line utilities subprocedure on the system containing the deployment manager profile.

    Note: If you are migrating from WebSphere Enterprise Service Bus version 6.0.2, you must use the Migrating a profile using the BPM migration command-line utilities procedure.

11. **Update Cloudscape or Derby cell scoped databases.**

If you are using Cloudscape or Derby databases for the Common Database, you must ensure that you are using the supported version. For instructions on migrating Cloudscape to Derby, see Migrating IBM Cloudscape or Apache Derby databases

12. **Copy the Common database migration and upgrade scripts to your database system.**

    On the target migration system, find the Common database migration and upgrade commands and scripts that are customized to your database type and copy them to your database system. The commands and scripts are located in the following directories: *install_root*/dbscripts/CommonDB/*database_type*

    Note: The Business Process Choreographer database commands and scripts are copied in a later process using the DBDesignGenerator command. For more information, see "Upgrading the Business Process Choreographer database schema" on page 75.
    Use the following table to determine the directory name that corresponds to your particular database type:

Database type	Directory name
DB2 Universal Database (for all operating systems except z/OS and i5/OS)	DB2
DB2 Universal Database for i5/OS	DB2iSeries
DB2 for z/OS Version 8.x	DB2zOSV8 - Use scripts in this directory if your initial database configuration used DB2 z/OS v8 (uses long table names) or if you upgraded from DB2 z/OS v7 to DB2 z/OS v8
DB2 for z/OS Version 9.x	DB2zOSV9 - Use scripts in this directory if your initial database configuration used DB2 z/OS v9 or later (uses long table names) or if you upgraded from DB2 z/OS v7 to DB2 z/OS v9.
Derby	Derby 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.
Informix	Informix
Oracle	Oracle
Microsoft SQL Server	SQLServer

13. **Upgrade the cell scoped Common database.**

    Upgrade the Common database schema manually using the Upgrading the Common database schema procedure if the database user that is defined for the Common database data source does not have sufficient privileges.

14. **Start the target deployment manager.**

    Start the target deployment manager using the startManager command from the *profile_root*/bin directory on the deployment manager system or from the deployment manager profile's First steps console.

Use the following syntax:

- Linux UNIX **On Linux and UNIX platforms:** `startManager.sh`
- Windows **On Windows platforms:** `startManager.bat`

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

15. **Update the data source configuration.** If you have data sources using embedded data direct driver and you did not update them in the source environment to use a licensed Data Direct JDBC driver or Microsoft JDBC driver, update the Data Source configuration. To do this, use the following procedure.

    Attention: The `SystemOut.log` file might reflect errors because some components could not establish connection to the database.

    a. Log in to the administrative console.
    b. 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.
    c. Delete the existing Data Source that uses embedded driver.
    d. Use the Test Connection option to check if the data source configuration works.
    e. Restart the deployment manager.

16. **Migrate the non-clustered managed nodes.**

    Repeat steps 15 through 25 for each non-clustered manage node that is a source of the migration.

17. **Stop the non-clustered managed node migration source servers.**

    Stop the migration source server using the `stopServer` command from the *profile_root*`/bin` directory on the migration source system or from the profile's First steps console.

    Use the following syntax:

    - Linux UNIX **On Linux and UNIX platforms:** `stopServer.sh` *server_name* `-username` *user_name* `-password` *password*
    - Windows **On Windows platforms:** `stopServer.bat` *server_name* `-username` *user_name* `-password` *password*

    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.

18. **Stop the non-clustered managed node migration source node agent.**

    Stop the migration source's node agent using the `stopNode` command from the *profile_root*`/bin` directory of the migration source system.

Use the following syntax:

- Linux UNIX **On Linux and UNIX platforms:** `stopNode.sh -username user_name -password password`
- Windows **On Windows platforms:** `stopNode.bat -username user_name -password password`

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 system.

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

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

19. **Backup the non-clustered managed node migration source profile.**

    Back up the profile configuration on the non-clustered managed node using the `backupConfig` command.

    Use the following syntax to backup a profile named profile1 to /ProfileBackups/profile1.zip.

    - Linux UNIX **On Linux and UNIX platforms:** `backupConfig.sh /ProfileBackups/profile1.zip -profileName profile1`
    - Windows **On Windows platforms:** `backupConfig.bat c:\ProfileBackups\profile1.zip -profileName profile1`

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

20. **Backup the server scoped product databases configured for the non-clustered managed node.**

    Backup the following product databases that are configured for the non-clustered managed node according to the documentation for your database:

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

21. **Migrate the non-clustered managed node.**

    The BPM profile migration wizard or the BPM migration command-line utilities can be used to migrate the non-clustered managed node source profile. To use the BPM profile migration wizard, follow the Migrating a profile using the BPM profile migration wizard sprocedure on the system containing the non-clustered managed node profile. To use the BPM migration command-line utilities, follow the Migrating a profile using the BPM migration command-line utilities procedure on the system containing the non-clustered managed node profile.

    Note: If you are migrating from WebSphere Enterprise Service Bus version 6.0.2, you must use the Migrating a profile using the BPM migration command-line utilities procedure.

22. **Update Cloudscape or Derby non-clustered managed node scoped databases.**

    If you are using Cloudscape or Derby databases configured for the non-clustered managed node scoped databases, you must ensure that you are using the supported version. For instructions on migrating Cloudscape to Derby, see Migrating IBM Cloudscape or Apache Derby databases

23. **Migrate the non-clustered managed node product databases.**

    Migrate each of the product databases that are configured on the non-clustered managed node using the following procedures:

    a. Upgrade the Business Process Choreographer Database schema manually using the Upgrading the Business Process Choreographer database schema procedure if either of the following conditions are true:
       - You did not use the default table spaces for the Business Process Choreographer database. If you used the sample Business Process Choreographer configuration, or have created all of the database objects in the default table spaces specified in the sample SQL scripts, your database uses the default table spaces. This is typically the case for a test environment.
       - The database user that is configured for the BPEDB data source is not authorized to perform all of the following operations: create and alter tables, create and drop indexes and views, and for the table SCHEMA_VERSION: query, update, delete, and insert.

    b. Migrate the Business Process Choreographer database data using the Migrating the Business Process Choreographer database data procedure if your source version you are migrating from is 6.0.2, 6.1.0, or 6.1.2.

    c. Migrate the Business Space database schema using the Migrating the Business Space database schema procedure.

    d. Migrate the Business Space database data using the Migrating the Business Space Database data procedure.

    e. Optional: Migrate the messaging engine database if it is needed for your environment. To learn more about when and how to migrate the messaging engine, see Migrating a messaging engine based on a data store on the WebSphere Application Server Version 7.0 information center.

24. **Optional: Migrate the Business Rules Manager.**

    The Business Rules Manager is automatically migrated when the last node in the cell is migrated but if the migrated non-clustered managed node contains the Business Rules Manager it can manually be migrated.

    To manually migrate the Business Rules Manager for the server server1 and the non-clustered managed node node1 use the following command:

    ```
    wsadmin -f installBRManager.jacl -s server1 -n node1
    ```

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

25. **Enable synchronization for the non-clustered managed node.**

    Enable synchronization for the non-clustered managed node that has been migrated using administrative console on the target deployment manager.

    Go to System Administration-> Node agents.

    Click on the nodeagent for the node.

    Click on the "File synchronization server".

    Restore the settings for "Enable service at server startup", "Automatic Migration" and "Startup synchronization".

Click Apply, Ok and Save to save the configuration changes.

26. **Start the migration target non-clustered managed node agent.**

    Start the migration target non-clustered managed node 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.

    Use the following syntax:

    - Linux UNIX **On Linux and UNIX platforms:** `startNode.sh`
    - Windows **On Windows platforms:** `startNode.bat`

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

27. **Start the migration target non-clustered managed node server.**

    Start the migration target non-clustered managed node 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.

    Use the following syntax:

    - Linux UNIX **On Linux and UNIX platforms:** `startServer.sh` *server_name*
    - Windows **On Windows platforms:** `startServer.bat` *server_name*

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

28. **Migrate the clusters.**

    Repeat steps 27 through 45 for each cluster in the network deployment environment that needs to be migrated.

    Divide the nodes that contain servers that contribute to the cluster into two roughly equivalent sized groups, group A and group B. The group B nodes will continue to service consumer requests while the group A nodes are taken offline and migrated. When the group A nodes are migrated, all nodes will be stopped, the databases configured for the cluster will be migrated, and the migrated group A nodes will be started and can begin servicing consumer requests. The group B nodes will then be migrated and started. Staggering the migration over the two groups of nodes will minimize the amount of time the cluster will need to be down in order to migrate the product databases.

29. **Stop the group A clustered managed node migration source servers.**

    Repeat this step for each server associated with a clustered managed node that will be migrated as part of group A.

    Stop the migration source server using the `stopServer` command from the *profile_root*`/bin` directory on the migration source system or from the profile's First steps console.

    Use the following syntax:

    - Linux UNIX **On Linux and UNIX platforms:** `stopServer.sh` *server_name* `-username` *user_name* `-password` *password*
    - Windows **On Windows platforms:** `stopServer.bat` *server_name* `-username` *user_name* `-password` *password*

    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.

30. **Stop the group A clustered managed node migration source node agents.**

    Repeat this step for each node agent associated with a clustered managed node that will be migrated as part of group A.

    Repeat this step for each node agent that is impacted by the migration.

    Stop the migration source's node agent using the `stopNode` command from the *profile_root*`/bin` directory of the migration source system.

    Use the following syntax:

    - Linux UNIX **On Linux and UNIX platforms:** `stopNode.sh -username` *user_name* `-password` *password*
    - Windows **On Windows platforms:** `stopNode.bat -username` *user_name* `-password` *password*

    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 system.

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

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

31. **Backup the group A migration source profiles.**

    Repeat this step for each profile that will be migrated in group A.

    Back up the profile configuration on the non-clustered managed node using the `backupConfig` command.

    Use the following syntax to backup a profile named profile1 to /ProfileBackups/profile1.zip.

    - Linux UNIX **On Linux and UNIX platforms:** `backupConfig.sh /ProfileBackups/profile1.zip -profileName profile1`
    - Windows **On Windows platforms:** `backupConfig.bat c:\ProfileBackups\profile1.zip -profileName profile1`

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

32. **Migrate the group A managed nodes.**

    This step should be repeated for each group A managed node in the cluster.

    The BPM profile migration wizard or the BPM migration command-line utilities can be used to migrate the clustered managed node source profile. To use the BPM profile migration wizard, follow the Migrating a profile using the

BPM profile migration wizard subprocedure on the system containing the clustered managed node profile. To use the BPM migration command-line utilities, follow the Migrating a profile using the BPM migration command-line utilities subprocedure on the system containing the clustered managed node profile.

Note: If you are migrating from WebSphere Enterprise Service Bus version 6.0.2, you must use the Migrating a profile using the BPM migration command-line utilities procedure.

33. **Stop the group B clustered managed node migration source servers.**

    Repeat this step for each server associated with a clustered managed node that will be migrated as part of group B.

    Stop the migration source server using the `stopServer` command from the *profile_root*`/bin` directory on the migration source system or from the profile's First steps console.

    Use the following syntax:

    - Linux UNIX **On Linux and UNIX platforms:** `stopServer.sh` *server_name* `-username` *user_name* `-password` *password*
    - Windows **On Windows platforms:** `stopServer.bat` *server_name* `-username` *user_name* `-password` *password*

    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.

34. **Stop the group B clustered managed node migration source node agents.**

    Repeat this step for each node agent associated with a clustered managed node that will be migrated as part of group B.

    Repeat this step for each node agent that is impacted by the migration.

    Stop the migration source's node agent using the `stopNode` command from the *profile_root*`/bin` directory of the migration source system.

    Use the following syntax:

    - Linux UNIX **On Linux and UNIX platforms:** `stopNode.sh` `-username` *user_name* `-password` *password*
    - Windows **On Windows platforms:** `stopNode.bat -username` *user_name* `-password` *password*

    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 system.

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

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

35. **Migrate the cluster.**

    Migrate the cluster scoped profile using the `BPMMigrateCluster` command from the *`install_root`*`/bin` directory on the system containing the deployment manager.

    Use the following syntax to migrate a cluster named applicationCluster1 with a deployment manager profile named dmgrProfile copied in the /MigrationSnapshots/ProcServer620 directory:

    - Linux UNIX **On Linux and UNIX platforms:** `BPMMigrateCluster.sh /MigrationSnapshots/ProcServer620 applicationCluster1 dmgrProfile`
    - Windows **On Windows platforms:** `BPMMigrateCluster.bat c:\MigrationSnapshots\ProcServer620 applicationCluster1 dmgrProfile`

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

36. **Enable synchronization for all clustered nodes.**

    Enable synchronization for all nodes in the cluster (both group A and group B) using the administrative console on the target deployment manager.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.

37. **Back up the group A migration source profiles.**

    Repeat this step for each profile that will be migrated in group A. This backup is needed in case the next step for running the syncNode command fails. After resolving the issue with syncNode, you can restore the backup before running syncNode command again.

    Back up the profile configuration on the non-clustered managed node using the `backupConfig` command.

    Use the following syntax to backup a profile named profile1 to /ProfileBackups/profile1.zip.

    - Linux UNIX **On Linux and UNIX platforms:** `backupConfig.sh /ProfileBackups/profile1.zip -profileName profile1`
    - Windows **On Windows platforms:** `backupConfig.bat c:\ProfileBackups\profile1.zip -profileName profile1`

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

38. **Synchronize all Group A nodes.**

    Repeat this step for each group A clustered managed node in the cluster.

    Synchronize the node with the target deployment manager using syncNode command from the *`profile_root`*`/bin` directory of the migration target profile or from the target profile's First steps console.

    Use the following syntax:

- Linux UNIX **On Linux and UNIX platforms:** `syncNode.sh` *deployment_manager_machine_name_or_ip_address deployment_manager_port_no*
- Windows **On Windows platforms:** `syncNode.bat` *deployment_manager_machine_name_or_ip_address deployment_manager_port_no*

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

39. **Back up the cluster scoped product databases configured for the cluster.**

    Backup the following product databases that are configured for the cluster according to the documentation for your database:

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

40. **Update Cloudscape or Derby cluster scoped databases.**

    If you are using Cloudscape or Derby databases configured for the cluster, you must ensure that you are using the supported version. For instructions on migrating Cloudscape to Derby, see Migrating IBM Cloudscape or Apache Derby databases

41. **Migrate the cluster scoped product databases.**

    Migrate each of the product databases that are configured for the cluster using the following procedures:

    a. Upgrade the Business Process Choreographer Database schema manually using the Upgrading the Business Process Choreographer database schema procedure if either of the following conditions are true:
       - You did not use the default table spaces for the Business Process Choreographer database. If you used the sample Business Process Choreographer configuration, or have created all of the database objects in the default table spaces specified in the sample SQL scripts, your database uses the default table spaces. This is typically the case for a test environment.
       - The database user that is configured for the BPEDB data source is not authorized to perform all of the following operations: create and alter tables, create and drop indexes and views, and for the table SCHEMA_VERSION: query, update, delete, and insert.

    b. Migrate the Business Process Choreographer database data using the Migrating the Business Process Choreographer database data procedure if your source version you are migrating from is 6.0.2, 6.1.0, or 6.1.2.

    c. Migrate the Business Space database schema using the Migrating the Business Space database schema procedure.

    d. Migrate the Business Space database data using the Migrating the Business Space Database data procedure.

    e. Optional: Migrate the messaging engine database if it is needed for your environment. To learn more about when and how to migrate the messaging engine, see Migrating a messaging engine based on a data store on the WebSphere Application Server Version 7.0 information center.

42. **Optional: Migrate the Business Rules Manager.**

The Business Rules Manager is automatically migrated when the last node in the cell is migrated but if the migrated cluster contains the Business Rules Manager it can manually be migrated.

To manually migrate the Business Rules Manager for the cluster cluster1 use the following command:

```
wsadmin -f installBRManager.jacl -cl cluster1
```

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

43. **Start the group A migration target node agent.**

    Repeat this step for each group A clustered managed node in the cluster.

    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.

    Use the following syntax:

    - Linux UNIX **On Linux and UNIX platforms:** `startNode.sh`
    - Windows **On Windows platforms:** `startNode.bat`

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

44. **Start the group A migration target servers.**

    Repeat this step for each server associated with a group A clustered managed node in the cluster.

    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.

    Use the following syntax:

    - Linux UNIX **On Linux and UNIX platforms:** `startServer.sh` *server_name*
    - Windows **On Windows platforms:** `startServer.bat` *server_name*

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

45. **Backup the group B migration source profiles.**

    Repeat this step for each profile that will be migrated in group B.

    Back up the profile configuration on the non-clustered managed node using the `backupConfig` command.

    Use the following syntax to backup a profile named profile1 to /ProfileBackups/profile1.zip.

    - Linux UNIX **On Linux and UNIX platforms:** `backupConfig.sh /ProfileBackups/profile1.zip -profileName profile1`
    - Windows **On Windows platforms:** `backupConfig.bat c:\ProfileBackups\profile1.zip -profileName profile1`

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

46. **Migrate the group B managed nodes.**

    This step should be repeated for each group B managed node in the cluster.

The BPM profile migration wizard or the BPM migration command-line utilities can be used to migrate the managed node source profile.

The BPM profile migration wizard or the BPM migration command-line utilities can be used to migrate the clustered managed node source profile. To use the BPM profile migration wizard, follow the Migrating a profile using the BPM profile migration wizard subprocedure on the system containing the clustered managed node profile. To use the BPM migration command-line utilities, follow the Migrating a profile using the BPM migration command-line utilities subprocedure on the system containing the clustered managed node profile.

Note: If you are migrating from WebSphere Enterprise Service Bus version 6.0.2, you must use the Migrating a profile using the BPM migration command-line utilities procedure.

47. **Start the group B migration target node agent.**

    Repeat this step for each group B clustered managed node in the cluster.

    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.

    Use the following syntax:

    - Linux UNIX **On Linux and UNIX platforms:** `startNode.sh`
    - Windows **On Windows platforms:** `startNode.bat`

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

48. **Start the group B migration target servers.**

    Repeat this step for each server associated with a group B clustered managed node in the cluster.

    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.

    Use the following syntax:

    - Linux UNIX **On Linux and UNIX platforms:** `startServer.sh` *server_name*
    - Windows **On Windows platforms:** `startServer.bat` *server_name*

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

49. **Optional: Uninstall the source deployment manager.**

    Once the migration is complete, the migration source deployment manager can be uninstalled.

50. **Remove Compatibility Mode.**

    If you chose the compatibility option (which is the default), and if all your nodes are completely migrated to the target version, run the `convertScriptCompatibility` script from the *install_root*`/bin` directory on the deployment to remove compatibility.

    Use the following syntax:

    - Linux UNIX **On Linux and UNIX platforms:** `convertScriptCompatibility.sh`

- Windows **On Windows platforms:** `convertScriptCompatibility.bat`

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

Results

The network deployment environment is migrated to the target version.

What to do next

Verify that the migration was successful. For instructions, see "Verifying migration" on page 86.

Runtime migration subprocedures

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

Migrating a profile using the BPM profile migration wizard

The BPM profile migration wizard is a graphical user interface (GUI) that guides you through the process of migrating a profile. Migrating a profile is just one step in a series of steps required to migrate a stand-alone or network deployment environment.

Before you begin

Make sure you have performed the required steps that precede the step for invoking the migration wizard. These steps differ depending on whether you are migrating a stand-alone environment or a network deployment environment. See the Migrating a stand-alone environment, Migrating a network deployment environment with full downtime, or Migrating a network deployment environment with minimal downtime topics.

About this task

This procedures describes the steps necessary to use the BPM profile migration wizard to migrate a profile.

Procedure

1. **Invoke the migration wizard.**

   Invoke the migration wizard using the `BPMMigrate` command from the *`target_install_root`*`/bin` directory.

   Use the following syntax:

   - Linux UNIX **On Linux and UNIX platforms:** `BPMMigrate.sh`
   - Windows **On Windows platforms:** `BPMMigrate.bat`

   For more information about the `BPMMigrate` command, see the BPMMigrate command-line utility topic.
2. **Read the Welcome screen.**

   On the Business Process Management Profile Migration Wizard Welcome screen, read the information on the panel to learn about the migration process, and then click **Next**.
3. **Select the wizard migration type: Typical or Custom.**

On the Select Typical or Custom migration screen, select either a typical wizard migration or a custom wizard migration, and then click **Next**.

- If you select **Typical**, the migration wizard migrates the BPM profile with the default configuration settings.
- If you select **Custom**, the migration wizard allows you to customize the configuration settings.

The default configuration settings are:

- **Snapshot directory**:
  - Linux UNIX `/MigrationSnapshots/source_install_root`
  - Windows `C:\MigrationSnapshots\source_install_root`
- **Target profile name**: The default for the target profile name is the source profile name
- **Target profile directory**: The default for the target profile directory is the *target_install_directory*/profiles/*source_profile_name* where *source_profile_name* is the name of the source profile
- **Port value assignments**: Same as source profile port assignments
- **Script compatibility (deployment manage profiles only)**: Set to true, so scripts from the source profile are still available post migration
- **Application directory settings (deployment manager profiles only)**: Default target installation directory of the target profile

4. **Select the source installation.**

   On the Select an installation to use as the source of the migration screen, select the source installation directory from the list of detected BPM products or select **Browse** to select the installation directory of BPM products not detected, and then click **Next**.

   Restriction: If you are migrating from WebSphere ESB version 6.0.2.x, you must use the "Migrating a profile using the BPM command-line utilities" on page 67 procedure.

5. **Select the source profile.**

   In the Select a source profile to use as the source of the migration screen, select the source profile from the list, enter the username and password if the profile has security enabled, then click **Next**.

6. **Define the custom settings or skip to the verify step for a typical migration.**

   Note: If you selected **Typical** in Step 3 on page 63, skip to Step 7 on page 66.

   If you selected **Custom** in Step 3 on page 63, use the following steps.

   a. **Select the snapshot directory.**

      On the Enter or browse for the snapshot directory to use for the source profile screen, keep the default snapshot directory or click **Browse** to navigate to a new snapshot directory, then click **Next**.

   b. **Specify the target profile name and target profile directory.**

      On the Select the target profile name and directory screen, keep the default target profile name and directory, or enter a new target profile name and directory in the **Target profile name** and **Target profile directory** fields, then click **Next**.

   c. **Select the application migration setting.**

      Note: This screen appears only if you are migrating a deployment manager profile.

On the Select the application migration setting screen, specify where the migrated applications should be located and then click **Next**. The default selection is: **Install the applications in the default directory of the target installation**.

- **Install the applications in the default directory of the target installation**.
- **Keep the current application installation directories**.

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

  – Mixed-node support limitations must be followed. This means that the following support cannot be used when invoking the wsadmin command:
    - Precompile JSP
    - Use Binary Configuration
    - Deploy EJB
  – You risk losing the migrated applications unintentionally if you later delete applications from these locations when administering (uninstalling for example) your previous installation.

d. **Select the port migration setting.**

   Note: This screen appears only if you are migrating a stand-alone profile.

   On the Select the port migration setting screen, select one of the following options for assigning target profile port values, and then click **Next**.

   - **Use the same port assignments as the source profile.**
   - **Do not override the ports that were created with the target profile.**
   - **Assign available ports to the target profile beginning with the following port number:**

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

   Note: The default selection is: **Use the same port assignments as the source profile.**

e. **Select the script compatibility setting.**

   Note: This screen appears only if you are migrating a deployment manager profile.

   On the Select the script compatibility setting screen, select or clear the **Enable source profile administrative scripts for use in the target installation** box, then click **Next**. Selecting this option sets the optional WebSphere Application Server `-scriptCompatibility` parameter to true. Setting this parameter to true enables the migration to create the following WebSphere Application Server version 6.x configuration definitions:

   - Transport
   - ProcessDef
   - Version 6.x SSL

   instead of the following WebSphere Application Server version 7.0 configuration definitions:

   - Channels

- ProcessDefs
- Version 7.0 SSL

Select this option in order to minimize impacts to existing administration scripts. For example, if you have existing **wsadmin** scripts or programs that use third-party configuration APIs to create or modify the version 6.x configuration definitions, select this option.

Note: This is meant to provide a temporary transition until all of the nodes in the environment are at the WebSphere Application Server version 7.0 level. When they are all at version 7.0, you should perform the following actions:

1) Modify your administration scripts to use all of the version 7.0 settings
2) Use the convertScriptCompatibility command to convert your configurations to match all of the version 7.0.

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

7. **Verify the migration wizard selections.**

   On the Profile migration summary screen, verify the migration selections you made in the wizard, then click **Next** to begin the migration.

8. **Monitor the status of the migration.**

   The Migration execution screen displays the status of the profile migration. Monitor the migration to validate that it is running successfully.

9. **Retry the migration if it fails.**

   If the profile migration fails while copying the source profile, creating the target profile, or migrating the source profile to the target profile, use the following procedure to retry the migration.

   a. Fix the root cause of the failure.
   b. Remove the following artifacts created by the failed migration:
      - The snapshot directory
      - The target profile (using the manageprofiles command-line utility).

      Note: If a deployment manager profile was being migrated, and the source deployment manager has been disabled, it should be re-enabled using the `migrationDisablementReversal` command to rollback the migration. However, if the profile migration is going to be re-executed, reversing the disablement of the deployment manager is unnecessary.
   c. Use the back button or restart the wizard to run the migration again.

10. Click **Next** if the migration completed successfully, and click **Finish** to exit the wizard.

Results

The profile is migrated from an earlier version of a BPM product to WebSphere Process Server version 7.0.

What to do next

Verify that the migration was successful. For instructions, see “Verifying migration” on page 86.

Migrating a profile using the BPM command-line utilities

Use this subprocedure for migrating a profile using the command-line utilities

Before you begin

See the Migrating a stand-alone environment, Migrating a network deployment environment with full downtime, and Migrating a network deployment environment with minimal downtime topics.

About this task

Follow these steps to migrate a profile using the command line utilities.

Procedure

1. **Create a copy of the source profile.**

   Create a copy of the configuration files in the source profile that will be migrated to the target profile using the `BPMSnapshotSourceProfile` command from the *`install_root`*`/bin` directory. The user-specified snapshot directory should not be located in the source or target product installation directories, so those directories can be removed later if necessary without impacting the configuration files in the snapshot directory.

   Use the following syntax to copy a source profile named sourceProfile1 located in the ProcServer620 installation root directory to the `/MigrationSnapshots/ProcServer620 snapshot` directory:

   - Linux UNIX **On Linux and UNIX platforms:**
     ```
     BPMSnapshotSourceProfile.sh /opt/ibm/WebSphere/ProcServer620
     sourceProfile1 /MigrationSnapshots/ProcServer620
     ```
   - Windows **On Windows platforms:**
     ```
     BPMSnapshotSourceProfile.bat
     "C:\Program Files\IBM\WebSphere\ProcServer620" sourceProfile1
     c:\MigrationSnapshots\ProcServer620
     ```

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

2. **Create the target profile.**

   If you are migrating a profile from any product and source combination **other** than WebSphere ESB version 6.0.2, create the target profile using the `BPMCreateTargetProfile` command from the *`install_root`*`/bin` directory. This profile will not be ready for use until the `BPMMigrateProfile` command is used to migrate the source profile to the new target profile.

   Use the following syntax to create a target profile for the migration using the source profile named sourceProfile1 copied to the `/MigrationSnapshots/ProcServer620` snapshot directory.

   - Linux UNIX **On Linux and UNIX platforms:**
     ```
     BPMCreateTargetProfile.sh /MigrationSnapshots/ProcServer620
     sourceProfile1
     ```
   - Windows **On Windows platforms:**
     ```
     BPMCreateTargetProfile.bat
     "C:\MigrationSnapshots\ProcServer620" sourceProfile1
     ```

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

   If you are migrating a WebSphere ESB profile from 6.0.2, you must use the Profile Management Tool or the manageprofiles command-line utility to create the target migration profile. For more information, see Creating profiles.

3. **Migrate the source profile to the target profile.**

   Migrate the source profile to the target profile using the `BPMMigrateProfile` command. This command reads the configuration information from the snapshot directory specified by the `BPMSnapshotSourceProfile` command and migrates it to the target profile.

   Use the following syntax to migrate the source profile named sourceProfile1 copied into the `/MigrationSnapshots/ProcServer620` directory to the target profile:

   - Linux UNIX **On Linux and UNIX platforms:** `BPMMigrateProfile.sh /MigrationSnapshots/ProcServer620 sourceProfile1`
   - Windows **On Windows platforms:** `BPMMigrateProfile.bat C:\MigrationSnapshots\ProcServer620 sourceProfile1`

   If the source profile has security enabled, the `-username` and `-password` parameters are required and the user name provided must be a member of the operator or administrator role.

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

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

4. **Check the migration status.**

   Use the `BPMMigrationStatus` command to verify the current state of the migration.

   Use the following syntax:

   - Linux UNIX **On Linux and UNIX platforms:** `BPMMigrationStatus.sh`
   - Windows **On Windows platforms:** `BPMMigrationStatus.bat`

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

Results

The profile is migrated from an earlier version of WebSphere Process Server to WebSphere Process Server version 7.0.

What to do next

Verify that the migration was successful. For instructions, see “Verifying migration” on page 86.

Migrating a stand-alone profile to a remote system

A stand-alone server migration subprocedure for migrating a profile to a remote system.

Before you begin

See the Migrating a stand-alone server profile topic.

About this task

Follow the steps in this procedure to migrate a profile to a remote system.

Procedure

1. **Create a default profile on the target system.** On the migration target system, create a default profile. Use the instructions in the following topic to create the default target profile: Creating profiles.
2. **Create the remote migration utilities image.**

   On the migration target system, or any system that has version 7.0 installed, create a remote migration image using the `BPMCreateRemoteMigrationUtilities` command from the *`install_root`*`/bin` directory.

   Use the following syntax:

   - Linux UNIX **On Linux and UNIX platforms:** `BPMCreateRemoteMigrationUtilities.sh remoteMigrationUtilities.gzip`
   - Windows **On Windows platforms:** `BPMCreateRemoteMigrationUtilities.bat remoteMigrationUtilities.zip`

   For more information about the `BPMCreateRemoteMigrationUtilities` command, see the BPMCreateRemoteMigrationUtilities command topic.
3. **Copy the remote migration utilities to the source system.**

   Using ftp, rcp, or some other mechanism, copy the remote migration utilities from the target system to the source system and unzip the remote migration utilities on the source system into their own unique directory.
4. **Snapshot the migration source profile.**

   On the migration source system use the `BPMSnapshotSourceProfile` command from the remote migration utilities `bin` directory to create a snapshot directory containing the configuration files that will be migrated.

   Use the following syntax to snapshot a source profile named sourceProfile1 located in the ProcServer620 installation root directory to the `/MigrationSnapshots/ProcServer620` snapshot directory:

   - Linux UNIX **On Linux and UNIX platforms:**
     ```
     BPMSnapshotSourceProfile.sh /opt/ibm/WebSphere/ProcServer620
     sourceProfile1 /MigrationSnapshots/ProcServer620
     ```
   - Windows **On Windows platforms:**
     ```
     BPMSnapshotSourceProfile.bat
     "C:\Program Files\IBM\WebSphere\ProcServer620" sourceProfile1
     c:\MigrationSnapshots\ProcServer620
     ```

   For more information about the `BPMSnapshotSourceProfile` command, see the BPMSnapshotSourceProfile command topic.
5. **Copy the migration source snapshot directory to the migration target system.**

   Create a zip of the source snapshot directory, copy it to the same directory on the target system, and unzip it there.
6. **Create the target profile.**

   If you are migrating a profile from any product and source combination **other** than WebSphere ESB version 6.0.2, create the target profile using the `BPMCreateTargetProfile` command. This profile will not be ready for use until the `BPMMigrateProfile` command is used to migrate the source profile to the new target profile.

   Use the following syntax to create a target profile for the migration using the source profile named sourceProfile1 copied to the `/MigrationSnapshots/ProcServer620` snapshot directory.

   - Linux UNIX **On Linux and UNIX platforms:**
     ```
     BPMCreateTargetProfile.sh /MigrationSnapshots/ProcServer620
     sourceProfile1
     ```

- Windows **On Windows platforms:** `BPMCreateTargetProfile.bat -remoteMigration true C:\MigrationSnapshots\ProcServer620 sourceProfile1`

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

If you are migrating a WebSphere ESB profile from 6.0.2, you must use the Profile Management Tool or the manageprofiles command-line utility to create the target migration profile.

7. **Migrate the source profile to the target profile.**

Migrate the source profile to the target profile using the `BPMMigrateProfile` command. This command will read the configuration information from the snapshot directory specified by the `BPMSnapshotSourceProfile` command and copied over to the target system and migrate it to the target profile.

Use the following syntax to migrate the source profile named sourceProfile1 copied into the `/MigrationSnapshots/ProcServer620` directory to the target profile:

- Linux UNIX **On Linux and UNIX platforms:** `BPMMigrateProfile.sh /MigrationSnapshots/ProcServer620 sourceProfile1`
- Windows **On Windows platforms:** `BPMMigrateProfile.bat C:\MigrationSnapshots\ProcServer620 sourceProfile1`

If the source profile does not have security enabled the username and password parameters are unnecessary, otherwise, the user name provided must be a member of the operator or administrator role.

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

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

8. **Check the migration status.**

Use the `BPMMigrationStatus` command to verify the current state of the migration.

- Linux UNIX **On Linux and UNIX platforms:** `BPMMigrationStatus.sh`
- Windows **On Windows platforms:** `BPMMigrationStatus.bat`

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

9. Scan the file system under the profile directory for occurrences of old hostname value. Analyze the configuration where the old hostname is still being used and replace it with new hostname, unless the old hostname is needed, such as of the database is still present on the old hostname machine.

Results

The profile is migrated from an earlier version of a BPM product to WebSphere Process Server version 7.0 on a remote system.

What to do next

Verify that the migration was successful. For instructions, see “Verifying migration” on page 86.

Migrating a stand-alone server while upgrading an operating system

A stand-alone server migration subprocedure for migrating a profile on a system whose operating system is being upgraded.

Before you begin

See the Migrating a stand-alone server profile topic.

About this task

Follow the steps in this procedure to migrate a profile on a system whose operating system is being upgraded.

Procedure

1. **Create the remote migration utilities image.**

   From any system that has a version 7.0 installed, create a remote migration utilities image using the `BPMCreateRemoteMigrationUtilities` command from the *install_root*`/bin` directory.

   Use the following syntax:

   - Linux UNIX **On Linux and UNIX platforms:**
     ```
     BPMCreateRemoteMigrationUtilities.sh remoteMigrationUtilities.gzip
     ```
   - Windows **On Windows platforms:**
     ```
     BPMCreateRemoteMigrationUtilities.bat remoteMigrationUtilities.zip
     ```

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

2. **Copy the remote migration utilities to the source system.**

   Using ftp, rcp, or some other mechanism, copy the remote migration utilities from the target system to the source system and unzip the remote migration utilities on the source system into their own unique directory.

3. **Snapshot the migration source profile.**

   On the migration source system use the `BPMSnapshotSourceProfile` command from the remote migration utilities bin directory to create a snapshot directory containing the configuration files that will be migrated.

   Use the following syntax to snapshot a source profile named sourceProfile1 located in the ProcServer620 installation root directory to the `/MigrationSnapshots/ProcServer620` snapshot directory:

   - Linux UNIX **On Linux and UNIX platforms:**
     ```
     BPMSnapshotSourceProfile.sh /opt/ibm/WebSphere/ProcServer620
     sourceProfile1 /MigrationSnapshots/ProcServer620
     ```
   - Windows **On Windows platforms:** `BPMSnapshotSourceProfile.bat "C:\Program Files\IBM\WebSphere\ProcServer620" sourceProfile1 c:\MigrationSnapshots\ProcServer620`

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

4. **Copy the migration source snapshot directory to a temporary location.**

   Create a zip of the source snapshot directory, copy it to a remote system temporarily while the source system is being upgraded.

5. **Upgrade the source system's operating system.**

   Upgrade the system's operating system to the appropriate version.

6. **Install the migration target product(s).**

   Install the target product and latest fix packs on the same system as the source product of the migration.

   Note: To migrate from source profiles augmented by multiple products, the new version of those products must be installed into the same target installation directory. For example, if the source profile is augmented by WebSphere Process Server and WebSphere Business Monitor, both of those products must be installed into the same target installation directory.

7. **Restore the migration source snapshot directory.**

   Copy the snapshot directory zip file that was stored temporarily on the remote system back to the freshly upgraded target migration system.

8. **Create the target profile.**

   If you are migrating a profile from any product and source combination **other** than WebSphere ESB version 6.0.2, create the target profile using the `BPMCreateTargetProfile` command. This profile will not be ready for use until the `BPMMigrateProfile` command is used to migrate the source profile to the new target profile.

   Use the following syntax to create a target profile for the migration using the source profile named sourceProfile1 copied to the /MigrationSnapshots/ProcServer620 snapshot directory.

   - Linux UNIX **On Linux and UNIX platforms:**
     ```
     BPMCreateTargetProfile.sh /MigrationSnapshots/ProcServer620
     sourceProfile1
     ```
   - Windows **On Windows platforms:** `BPMCreateTargetProfile.bat "C:\MigrationSnapshots\ProcServer620" sourceProfile1`

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

   If you are migrating a WebSphere ESB profile from 6.0.2, you must use the Profile Management Tool or the manageprofiles command-line utility to create the target migration profile.

9. **Migrate the source profile to the target profile.**

   Migrate the source profile to the target profile using the `BPMMigrateProfile` command. This command will read the configuration information from the snapshot directory specified by the `BPMSnapshotSourceProfile` command and migrate it to the target profile.

   Use the following syntax to migrate the source profile named sourceProfile1 copied into the /MigrationSnapshots/ProcServer620 directory to the target profile:

   - Linux UNIX **On Linux and UNIX platforms:** `BPMMigrateProfile.sh /MigrationSnapshots/ProcServer620 sourceProfile1`
   - Windows **On Windows platforms:** `BPMMigrateProfile.bat "C:\MigrationSnapshots\ProcServer620" sourceProfile1`

   If the source profile does not have security enabled the `-username` and `-password` parameters are unnecessary, otherwise, the user name provided must be a member of the operator or administrator role.

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

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

10. **Check the migration status.**

    Use the `BPMMigrationStatus` command to verify the current state of the migration.

    - Linux UNIX **On Linux and UNIX platforms:** `BPMMigrationStatus.sh`
    - Windows **On Windows platforms:** `BPMMigrationStatus.bat`

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

Results

The profile is migrated from an earlier version of a BPM product to WebSphere Process Server version 7.0, and the operating system is upgraded.

What to do next

Verify that the migration was successful. For instructions, see "Verifying migration" on page 86.

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.

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 18.

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 a stand-alone environment, Migrating a network deployment environment with full downtime, and Migrating a network deployment environment with minimal downtime topics.

About this task

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

Database type	Directory name
DB2 Universal Database (for all operating systems except z/OS and i5/OS)	`DB2`
DB2 Universal Database for i5/OS	`DB2iSeries`

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.
Informix	`Informix`
Oracle	`Oracle`
Microsoft SQL Server	`SQLServer`

Procedure

Depending on your database type, use one of the following procedures to upgrade to a new database schema for the Common database.

- **DB2, Derby, Informix, Oracle, and SQLServer**

  For DB2, Derby, Informix, Oracle, and SQLServer, use the following procedure.

  Note: When you are migrating profiles with Oracle as database type, make sure you have view privileges before running upgrade scripts.

  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.

     Use the following syntax to run the command in interactive mode:

     – Linux UNIX **On Linux and UNIX platforms:** `upgradeSchema.sh`

     – Windows **On Windows platforms:** `upgradeSchema.bat`

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

- **DB2iSeries**

  For DB2iSeries, use the following procedure.

  1. Locate the Common database DB2iSeries SQL scripts on the database system that were copied from the following directory on the target migration system: *install_root*`/dbscripts/CommonDB/DB2iSeries`

The scripts that you will need to edit and then invoke have a file name that contains the source product version you are upgrading from (either 602, 610, 612, or 620), and begin with either **upgradeSchema** or **wbiserver_upgradeSchema**.

2. Check the SQL scripts, and modify them if necessary to meet your requirements. For example, you might need to change a user name, password, or file path.
3. Using your database client on your database system, connect to the database. This is to ensure that it can connect.

   Note: It is very important that your database system is fully set up to execute .sql scripts using database-specific tools. For example, if dbType is DB2_Universal, you can run any db2 command on the command prompt. The same is true for sqlplus commands for oracle and osql commands for SQL server.
4. From the directory containing the DB2iSeries SQL scripts on the database system, invoke each of the SQL scripts.

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 a stand-alone environment, Migrating a network deployment environment with full downtime, and Migrating a network deployment environment with minimal downtime 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.
      - Linux UNIX On Linux and UNIX platforms: *profile_root*/dbscripts/ProcessChoreographer/*database_type*/*database_name*/*database_schema*/createSchema.properties
      - Windows On Windows platforms: *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.

Database type	Directory name
DB2 Universal Database (for all operating systems except z/OS and i5/OS)	`DB2`
DB2 Universal Database for i5/OS	`DB2iSeries`
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.
Informix	`Informix`
Oracle	`Oracle`
Microsoft SQL Server	`SQLServer`

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)

- Linux UNIX On Linux and UNIX platforms, enter the following command:

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

- Windows On windows platforms, enter the following command:

```
install_root\util\dbUtils\DbDesignGenerator.bat
      -e copy_of_createSchema.properties_file
```

For more information about using this tool, see Creating the database design file using the database design tool.

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.

- Linux UNIX On Linux and UNIX platforms, enter the following command:

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

- Windows On windows platforms, 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 `createTablespaces`*`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 upgradeSchemas*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 77, 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 Linux, UNIX, Windows, or 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 79. 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 79.

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 a stand-alone environment, Migrating a network deployment environment with full downtime, and Migrating a network deployment environment with minimal downtime topics.

About this task

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

Database type	Directory name
DB2 Universal Database (for all operating systems except z/OS and i5/OS)	`DB2`
DB2 Universal Database for i5/OS	`DB2iSeries`
DB2 for z/OS Version 8.x	`DB2zOSV8` - Use scripts in this directory if your initial database configuration used DB2 z/OS v8 (uses long table names) or if you upgraded from DB2 z/OS v7 to DB2 z/OS v8
DB2 for z/OS Version 9.x	`DB2zOSV9` - Use scripts in this directory if your initial database configuration used DB2 z/OS v9 or later (uses long table names) or if you upgraded from DB2 z/OS v7 to DB2 z/OS v9.
Derby	`Derby` 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.
Informix	`Informix`
Oracle	`Oracle`
Microsoft SQL Server	`SQLServer`

Procedure

1. Log on to the database server as a user with read and write access on the database.
2. Connect to the database.
3. Locate the `migrateSchema` script in the profile you most recently configured, and save it to a location on the same system with the database.

   By default, the scripts are located in the following directory:

   - **Stand-alone server**: *profile_root*`/dbscripts/BusinessSpace/`*node_name_server_name*`/`*database_product_name*`/`*database_name*
   - **Cluster**: *profile_root*`/dbscripts/BusinessSpace/`*cluster_name*`/`*database_product_name*`/`*database_name*

   The script is located in the profile for the server or cluster that you most recently configured.

Note: This script may need to be modified if the default values do not match your environment.

Note: The SQL scripts *was_root*/dbscripts/BusinessSpace/*database_product_name*/*database_name* in can also be used to upgrade the Business Space database schema. These scripts must be modified to replace the variables *DB_NAME* and *DB_USER* with actual values, and the schema name must already exist in the database.

4. On the database system, invoke the migrateSchema command, using the following syntax:
   - Linux UNIX `migrateSchema.sh`
   - Windows `migrateSchema.bat`
   - **IBM i**: `migrateSchema`

   For more information about the migrateSchema command, see migrateSchema command-line utility for the Business Space database.
5. For DB2 and DB2 for z/OS, bind the command-line interface to the Business Space database using the following commands:

   ```
   db2 connect to database_name
   db2 bind DB2_installation_directory\bnd\@db2cli.lst blocking all
   grant public
   db2 connect reset
   ```

   where:

   database_name is the name of the Business Space database

   DB2_installation_directory is the directory where DB2 is installed
6. Start the server.

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.

     Use the following syntax:

     – Linux UNIX **On Linux and UNIX platforms:** startServer.sh *server_name*

     – Windows **On Windows platforms:** startServer.bat *server_name*

     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.

   Use the following syntax:

   – Linux UNIX **On Linux and UNIX platforms:** `startNode.sh`

   – Windows **On Windows platforms:** `startNode.bat`

   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.

   Use the following syntax:

   – Linux UNIX **On Linux and UNIX platforms:** `startServer.sh` *server_name*

   – Windows **On Windows platforms:** `startServer.bat` *server_name*

   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.

   Choose the script for your operating system:

   - **Windows**: `migrateBSpaceData.bat`
   - **AIX, HP-UX, Linux, Solaris**: `migrateBSpaceData.sh`

   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.

     Use the following syntax:

     – **On i5/OS platforms:** `stopServer` *server_name* `-username` *user_name* `-password` *password*

     – Linux UNIX **On Linux and UNIX platforms:** `stopServer.sh` *server_name* `-username` *user_name* `-password` *password*

     – Windows **On Windows platforms:** `stopServer.bat` *server_name* `-username` *user_name* `-password` *password*

     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.

     Use the following syntax:

     – Linux UNIX **On Linux and UNIX platforms:** `stopServer.sh` *server_name* `-username` *user_name* `-password` *password*

     – Windows **On Windows platforms:** `stopServer.bat` *server_name* `-username` *user_name* `-password` *password*

     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 migration wizard or 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. Check the profile's log files for fatal profile creation or augmentation errors. Profile log files are located in the following directory: *install_root*`/logs/manageprofiles`. The log files contain the profile name in them, for example:`create <profile name>.log`.
3. Check the server log files.
   a. Navigate to the *profile_root*`/logs/`*server_name* directory corresponding to the migrated profile.
   b. Review the `SystemOut.log` file and make sure there are no fatal errors.
   c. Review the `SystemErr.log` file and make sure there are no fatal errors.

4. 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.`
5. 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 105 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 6.2.0, 6.1.2, 6.1.0, or 6.0.2 *profile_root*`/bin` directory of the deployment manager that you need to roll back from version 7.0.

      Linux In a Linux environment, for example, use the following parameters:

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

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

      1) Go to the following directory:

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

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

2) If you see a `serverindex.xml_disabled` file in this directory, do the following:
   a) Delete or rename the `serverindex.xml` file.
   b) Rename the `serverindex.xml_disabled` file to `serverindex.xml`.

3. For each of the deployment cell's managed nodes that you need to roll back, do one of the following:
   a. If you backed up your previous managed node configuration using the **backupConfig** command or your own preferred backup utility, run the **restoreConfig** command or your own preferred utility to restore the version 6.2.0, 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.

      Linux In a Linux environment, for example, use the following parameters:

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

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

      1) Go to the following directory:

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

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

      2) If you see a `serverindex.xml_disabled` file in in this directory, do the following:
         a) Delete or rename the `serverindex.xml` file.
         b) Rename the `serverindex.xml_disabled` file to `serverindex.xml`.

4. Synchronize the managed nodes if they were ever running when the version 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 when following the steps in "Migrating a network deployment environment with minimal downtime" on page 48. 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 Network Deployment, version 6.1 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 Network Deployment, version 6.1 information center.
3. Migrate the managed node.

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

Important: If you do not have a backup copy of your version 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 87.

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 Network Deployment, version 6.1 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 Network Deployment, version 6.1 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*profile_root*`/bin` directory of the managed node.

        Linux In a Linux environment, for example, use the following parameters:

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

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

        1) Go to the following directory:

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

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

4. Start the version 7.0 deployment manager.
5. Synchronize the managed node.

   See Synchronizing nodes with the wsadmin tool in the WebSphere Application Server Network Deployment, version 6.1 information center.
6. If you chose to keep the installed applications in the same location as the prior release during migration to version 7.0 and any of the version 7.0 applications are not compatible with the prior release, install applications that are compatible.
7. Delete the version 7.0 managed profile.

   See Deleting a profile in the WebSphere Application Server Network Deployment, version 6.1 Information Center.
8. Start the rolled-back managed node in the version 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. In a network deployment environment, edit the file on the deployment manager.

         - Linux UNIX On Linux and UNIX platforms, the file is located in *`profile_root`*`/config/cells/`*`cellName`*`/wim/model`.
         - Windows On Windows platforms, the file is located in *`profile_root`*`\config\cells\`*`cellName`*`\wim\model`.

         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.

Linux UNIX **On Linux and UNIX platforms:**
Change to the directory *install_root*/ProcessChoreographer/config and enter the command:

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

Windows **On Windows platforms:**
Change to the directory *install_root*\ProcessChoreographer\config and enter the command:

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

Where *profileName* is the name of the deployment manager's profile and *clusterName* is the name of cluster where Business Process Choreographer is configured.

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.x, and you use DB2 for Linux, UNIX, Windows, or z/OS, to release the extra storage space used by the table space migration, delete the following old tables from the database:
   - PROCESS_TEMPLATE_B_O
   - ACTIVITY_TEMPLATE_B_O
   - SCOPED_VARIABLE_INSTANCE_B_O
   - CORRELATION_SET_INSTANCE_B_O
   - STAFF_QUERY_INSTANCE_O
   - TASK_TEMPLATE_O
   - TASK_INSTANCE_O

   Attention: Take care not to delete any of the new tables, which have similar names but have the suffix "_T".
7. Optional: Retune your database now or later. For example, for DB2 databases, run REORG and RUNSTATS.
8. If you 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 migration wizard

The BPM profile migration wizard is a graphical user interface (GUI) that guides you through the process of migrating a profile. The wizard is invoked by running the `BPMMigrate` command.

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

For more information about running the BPM profile migration wizard, see "Migrating a profile using the BPM profile migration wizard" on page 63.

BPM profile command-line tools

BPMMigrate

The `BPMMigrate` command invokes the BPM profile migration wizard that supports the migration of BPM profiles.

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

For more information about running the BPM profile migration wizard, see "Migrating a profile using the BPM profile migration wizard" on page 63.

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.

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

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.

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.

Profile creation error

While you are using the version 7.0 migration wizard to create a profile when migrating a configuration, you might see the following profile-creation error messages.

```
profileName: profileName cannot be empty
profilePath: Insufficient disk space
```

These error messages might be displayed if you enter a profile name that contains an incorrect character such as a space. Rerun the migration wizard, and verify that there are no incorrect characters in the profile name such as a space, quotation marks, or any other special characters.

Profile migration error

When you use the migration wizard to migrate a profile from WebSphere Process Server version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 to version 7.0 on a Solaris x64 processor-based system, the migration might fail during the BPMMigrateProfile step.

You might see messages similar to the following in *profile_root*`/logs/WASPostUpgrade.`*`time_stamp`*`.log`:

```
MIGR0327E: A failure occurred with stopNode.
MIGR0272E: The migration function cannot complete the command.
```

WebSphere Process Server version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 uses a Java virtual machine (JVM) in 32-bit mode. The migration wizard for WebSphere Process Server version 7.0 calls the `BPMMigrateProfile.sh` script, which attempts to run the JVM for version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 in the 64-bit mode when the server stops the version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 node.

Complete the following actions to remove the incomplete profile and enable WebSphere Process Server to correctly migrate the version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 profile:

1. On a command line, change to the *install_root*`/bin` directory.

   For example, type the following command:

   ```
   cd /opt/IBM/WebSphere/Procserver/bin
   ```

2. Locate the `BPMMigrateProfile.sh` script in the *install_root*`/bin` directory, and make a backup copy.
3. Open the `BPMMigrateProfile.sh` or `BPMMigrateProfile.bat` file in an editor, and perform the following actions:
   a. Locate the following line of code:

      UNIX Linux

      ```
      "$binDir" /setupCmdLine.sh
      ```

      Windows

      ```
      call "%~dp0setupCmdLine.bat" %*
      ```

   b. Insert the following line of code after the code that was identified in the previous step:

```
JVM_EXTRA_CMD_ARGS=""
```

c. Save the changes.

4. Repeat steps 2 through 4 with the`WASPostUpgrade.sh` or the `WASPostUpgrade.bat` file.
5. Delete the incomplete version 7.0 profile that was created during the migration process. Use the following procedure.
   a. Open a command prompt and run one of the following commands, based on your operating system:
      - Linux UNIX **On Linux and UNIX platforms:** `manageprofiles.sh -delete -profileName` *profile_name*
      - Windows **On Windows platforms:** `manageprofiles.bat -delete -profileName` *profile_name*

      The variable *profile_name* represents the name of the profile that you want to delete.
   b. Confirm that the profile deletion has completed by checking the following log file:
      - Linux UNIX **On Linux and UNIX platforms:** *install_root*`/logs/manageprofiles/`*profile_name*`_delete.log`
      - Windows **On Windows platforms:** *install_root*`\logs\manageprofiles\`*profile_name*`_delete.log`
6. Delete the *profile_root* directory of the version 7.0 profile that was removed in the previous step.
7. Rerun the migration wizard.

Servlet error

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

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

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

Deprecated command-line utility	Replacement command-line utility
`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`

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

Deprecated method	Recommended method for migration
`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**
`deleteCompletedProcess Instances.py` script parameters **-validFrom** and **-completedBefore**	Use these parameters: **-validFromUTC** and **-completedBeforeUTC**.
`deleteInvalidProcessTemplate.py` script parameter **-validFrom**	Use **-validFromUTC**.
`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

BOCopy Service, Two methods: copyInto() and copyIntoShallow()

The following two methods within the BOCopy Service are deprecated: `copyInto()` and `copyIntoShallow()`.

Doing both a copy and a set at the same time masks some of the problems you can encounter with the copy or the set. Because it is as easy to copy and set them separately, the workaround is simple. Use copy() then set() rather than using a combined version of the API.

Recommended migration action:

Use the following methods instead of `copyInto()` and `copyIntoShallow()`:

- Instead of `copyInto()`, use: `copy()` then `set()`
- Instead of `copyIntoShallow()`, use `copyShallow()` then `set()`

CEI parameters used for stand-alone profile which are common with the CommonDB

Most of the CEI parameters used for stand-alone profiles which are common with the CommonDB are deprecated.

Recommended migration action:

If you are using the `manageprofiles` command-line utility in version 6.2.0, 6.1.2, 6.1.0, or 6.0.2 and are planning to use the same command in version 7.0, it has to be modified to use the new set of parameters.

Note: If you are using Profile Management Tool, then the GUI (graphical user interface) takes care of passing the correct parameters.
The following table describes the CEI parameters that are being deprecated. Beginning with version 6.2, use the same parameters as those for CommonDB for the corresponding CEI parameters. Examples of how to modify the CEI parameters appear below the table.

Table 8. Deprecated CEI parameters

CEI variable name	CommonDB variable name	Applicable database
`nodeName`	`nodeName`	All
`ceiServerName`	`serverName`	All
`ceiDbExecuteScripts`	`dbDelayConfig`	All
`ceiJdbcClassPath`	`dbJDBCClasspath`	All
`ceiDbHostName`	`dbHostName`	All
`ceiDbPort`	`dbServerPort`	All
`ceiDbUser`	`dbUserId`	All except MSSQL
`ceiDbPassword`	`dbPassword`	All except MSSQL
`ceiOutputScriptDir`	`dbOutputscriptDir`	All
`ceiStorageGroup`	`dbStorageGroup`	DB2 z/OS
`ceiDbAliasName`	`cdbSchemaName`	DB2 z/OS
`ceiDbSubSystemName`	`dbConnectionLocation`	DB2 z/OS
`ceiNativeJdbcClassPath`	`dbJDBCClasspath`	DB2 iSeries® Native
`ceiCollection`	`cdbSchemaName`	DB2 iSeries Native
`ceiToolboxJdbcClassPath`	`dbJDBCClasspath`	DB2 iSeries Toolbox
`ceiCollection`	`cdbSchemaName`	DB2 iSeries Toolbox
`ceiDbInformixDir`	`dbLocation`	Informix
`ceiDbServerName`	`dbInstance`	Informix
`ceiDbSysUser`	`dbSysUserId`	Oracle
`ceiDbSysPassword`	`dbSysPassword`	Oracle

Example 1: manageprofiles

Here is an example which shows the old command and the new command if you are using the manageprofiles command-line utility. You no longer need to pass the CEI parameters except the "ceiDBName".

OLD

```
612 manageprofiles.bat -create -profileName -templatePath
\profileTemplates\default.wbiserver -dbType DB2_Universal -dbDelayConfig
false -dbCreateNew true -dbJDBCClasspath <classpath> -dbHostName localhost
-dbServerPort <port> -dbUserId <userid> -dbPassword <password>
-ceiDbProduct CEI_DB_DB2 -ceiDbExecuteScripts true -ceiJdbcClassPath
<classpath> -ceiDbHostName localhost -ceiDbPort <port> -ceiDbUser <userid>
-ceiDbPassword <password>
```

NEW

```
62 manageprofiles.bat -create -profileName -templatePath
\profileTemplates\default.wbiserver -dbDelayConfig false -dbType
DB2_Universal -dbJDBCClasspath <classpath> -dbHostName -dbServerPort <port>
-dbUserId <userid> -dbPassword <password>
```

"Everyone" value used to map Java EE roles BPEAPIUser and TaskAPIUser

The possible use of the value "Everyone" to map Java EE roles BPEAPIUser and TaskAPIUserd is deprecated.

Recommended migration action:

If you used the value "Everyone" to map Java EE roles BPEAPIUser and TaskAPIUser, fix your Business Process Choreographer client applications by logging in before using the Business Process Choreographer APIs.

FailedEventManagerMBean interface and API

The following FailedEventManagerMBean interface, methods, and operations are deprecated:

- `com.ibm.wbiserver.manualrecovery.FailedEventWithParameters` (whole class)
- `com.ibm.wbiserver.manualrecovery.FailedEventManager` (methods)
- `FailedEventManagerMBean.xml` (operations)

Recommended migration action:

It is recommended that you switch to the new interface and MBean operations only if custom code is used to manage failed events with FailedEventManagerMBean. The suggested new interfaces, methods, and operations are shown in the following table.

Table 9. New interfaces, methods, and operations forFailedEventManagerMBean

Deprecated interface, operation, or method	New interface, operation, or method
com.ibm.wbiserver.manualrecovery. FailedEventWithParameters	com.ibm.wbiserver.manualrecovery.SCAEvent
com.ibm.wbiserver.manualrecovery. FailedEventManager	

Table 9. New interfaces, methods, and operations forFailedEventManagerMBean (continued)

Deprecated interface, operation, or method	New interface, operation, or method
• List getFailedEventsForDestination(String destModuleName, String destComponentName, String destMethodName, int pagesize) throws FailedEventReadException; • List getFailedEventsForTimePeriod(Date begin, Date end, int pagesize) throws FailedEventReadException;	List<FailedEvent> queryFailedEvents(QueryFilters queryFilters, int offset, int maxRows) throws FailedEventReadException;
FailedEventWithParameters getFailedEventWithParameters(String msgId) throws FailedEventDataException;	SCAEvent getEventDetailForSCA(FailedEvent failedEvent) throws FailedEventDataException;
void discardFailedEvents(String[] msgIds) throws DiscardFailedException;	void discardFailedEvents(List<FailedEvent> failedEvents) throws DiscardFailedException;
void resubmitFailedEvents(String[] msgIds) throws ResubmissionFailedException;	void resubmitFailedEvents(List failedEvents) throws ResubmissionFailedException;
FailedEventManagerMBean.xml	
• getFailedEventsForDestination • getFailedEventsForTimePeriod	queryFailedEvents
getFailedEventWithParameters	getEventDetailForSCA
discardFailedEvents	discardFailedEvents with the following parameters: • name="failedEvents" • description="A list of failed events" • type="java.util.List"
resubmitFailedEvents	resubmitFailedEvents • name="failedEvents" • description="A list of failed events" • type="java.util.List"

WebSphere Connect JDBC Drivers (from DataDirect) for Microsoft SQL Server

The WebSphere Connect JDBC Drivers (from DataDirect) for Microsoft SQL Server that are shipped on the WebSphere Application Server Supplemental CDs are no longer shipped on the WebSphere Application Server, version 7, CDs.

Recommended migration action:

You must migrate any Microsoft SQL databases that are using the shipped DataDirect drivers to instead use the new JDBC driver that is provided by Microsoft. The new JDBC driver is not currently supported by WebSphere Process Server and WebSphere Enterprise Service Bus, but it will be supported in the future. You can either change to another database type (such as the Microsoft SQL embedded driver), or you can wait for the new JDBC driver to be supported by WebSphere Process Server and WebSphere Enterprise Service Bus, and migrate at that time.

Deprecated features in WebSphere Process Server version 6.1.2

WebSphere Process Server version 6.1.2 has no deprecated features.

Deprecated features in WebSphere Process server version 6.1

Container Manager Persistence over Anything (CMP/A)

The CMP/A support included with WebSphere Process Server is deprecated. This includes the runtime support for applications which have been customized to use CMP/A, the cmpdeploy.bat/.sh command line tool, and the following public APIs:

- com.ibm.websphere.rsadapter.WSProceduralPushDownHelper
- com.ibm.websphere.rsadapter.WSPushDownHelper
- com.ibm.websphere.rsadapter.WSPushDownHelperFactory
- com.ibm.websphere.rsadapter.WSRelationalPushDownHelper

Recommended migration action:

Convert CMP Entity Beans to use a relational data source, or have the CMP entity bean replaced by a different supported data persistence model.

You can also use WebSphere Adapters to replace your existing CMP/A applications. The Adapter tools use a 'Create, Retrieve, Update, and Delete' architecture for creating service interfaces that is very similar to the architecture that CMP/A uses.

JACL scripts (deprecated in WebSphere Application Server version 6.1)

JACL script files are deprecated in WebSphere Process Server to maintain consistency with the deprecation of JACL scripts in WebSphere Application Server.

Recommended migration action:

Use the corresponding .bat/.sh files or wsadmin commands to perform the same functions.

Note: The following Business Process Choreographer JACL scripts are not deprecated:

1. <*install_root*>\ProcessChoreographer\admin\bpcTemplates.jacl
2. <*install_root*>\ProcessChoreographer\config\bpeconfig.jacl
3. <*install_root*>\ProcessChoreographer\config\bpeunconfig.jacl
4. <*install_root*>\ProcessChoreographer\config\bpeupgrade.jacl
5. <*install_root*>\ProcessChoreographer\config\clientconfig.jacl

IBM Web Services Client for C++

The IBM Web Services Client for C++ is a 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:

  You can retrieve the staff activity associated with the inline human task for inline human tasks, and check the activity state using the getExecutionState() method on the ActivityInstanceData interface in the Generic Business Process EJB API.
- The field KIND_WPC_STAFF_ACTIVITY from interface Task is deprecated.

  Recommended migration action:

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

Deprecation of e-mail people assignment criteria

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

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

Recommended migration action:

E-mail addresses and the preferred language are resolved together with the user ID by the standard set of people assignment criteria with version 6.1. This deprecation information is thus especially important for those who write custom XSLT people

assignment criteria mapping (staff verb) files. If you do not intend to deploy version 6.0.2 task definitions, you do not need to support the deprecated people assignment criteria. Note that with version 6.1, the people assignment criteria "User Records by user ID" has been introduced and has to be supported by custom XSLT files, since it resolves e-mail addresses as fallback.

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

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

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

Recommended migration action:

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

Business Objects

The following Business Object methods are deprecated:

- com.ibm.websphere.bo.BOFactory.createByClass(java.lang.Class iterfaceClass);
- com.ibm.websphere.bo.BOType.getTypeByClass(java.lang.Class className);

Recommended migration action:

These methods will raise "function not supported" exceptions if they are called in version 6.1.

Common Event Infrastructure

Creation and editing of user-visible Common Base Events are deprecated.

Recommended migration action:

You can now use the tools to specify the Business Object data that is to be included in monitored emitted events.

zOS

The requirement to bind a String object into JNDI at esb/messageLogger/qualifier is deprecated.

Recommended migration action:

The Message Logger primitives will now store message information within the CommonDB database. Where necessary, during the profile augmentation phase, a WebSphere variable called ESB_MESSAGE_LOGGER_QUALIFIER will now be

created and its value set to that of the chosen CommonDB schema qualifier.

WebSphere InterChange Server

The APIs (application programming interfaces) listed in Supported WebSphere InterChange Server APIs are no longer deprecated.

Note: These APIs were formerly deprecated in WebSphere Process Server version 6.0.2.

Recommended migration action:

You should use these APIs only for applications with migrated WebSphere InterChange Server components. In all other cases, you should use the Service Data Objects for WebSphere Process Server.

WebSphere Enterprise Service Bus (WESB)

The current method to identify an SSL repertoire to be used when WESB communicates with a secured WSRR instance has been deprecated.

Recommended migration action:

A new property has been added to WSRR definitions to allow the specification of such a repertoire.

Deprecated features in WebSphere Process Server version 6.0.2

Human Task Manager

The task context variable %htm:task.clientDetailURL% is no longer required, and thus has been deprecated.

Recommended migration action:

No action is required.

The standard e-mail implementation used for all escalation e-mails in TEL has been deprecated, and replaced by native support for defining e-mails in TEL.

Recommended migration action:

Use the customizeable e-mail feature for escalations.

The following Task object methods that were deprecated in version 6.0 are no longer deprecated:

getInputMessageTypeName()
getOutputMessageTypeName()

Recommended migration action:

You can now use these methods.

Business Process Choreographer

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

Recommended migration action:

Use these corresponding methods:

- getProcessAdminTaskID() in combination with method getUsersInRole() of the HumanTaskManagerService interface, as follows:

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

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

  htm.getUsersInRole(procTemplData.getAdminTaskTemplateID(), WorkItem.REASON_ADMINISTRATOR)

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

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

Recommended migration action:

Use these corresponding methods:

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

The following JACL scripts are deprecated:

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

Recommended migration action:

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

SCA Admin Commands

The following commands (used with wsadmin) are deprecated:
- configSCAForServer
- configSCAForCluster

Recommended migration action:

Use these two commands in place of configSCAForServer for equivalent function:
- configSCAAsyncForServer
- [Optional; use only if required] configSCAJMSForServer

Use these two commands in place of configSCAForCluster for equivalent function:
- configSCAAsyncForCluster
- [Optional; use only if required] configSCAJMSForCluster

WebSphere InterChange Server

Note: These APIs are no longer deprecated in version 6.1.

The APIs (application programming interfaces) listed in Supported WebSphere InterChange Server APIs are deprecated.

Recommended migration action:

Code written for WebSphere Process Server should not use these interfaces.

IBM WebSphere InterChange Server Access for Enterprise JavaBeans™ (EJB) support is deprecated.

Recommended migration action:

Applications developed for use with WebSphere Process Server should not use Access for Enterprise JavaBeans.

Deprecated features in WebSphere Process Server version 6.0.1

WebSphere Process Server version 6.0.1 has no deprecated features.

Deprecated features in WebSphere Process Server version 6.0

Application programming model and container support features

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

Recommended migration action:

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

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

Recommended migration action:

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

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

Recommended migration action:

Use BPEL compensation for business processes.

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

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

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

    Replace with: getStoredQueryNames()
  - deleteWorkList(String workListName)

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

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

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

    not supported.
- The WorkListData object is deprecated.

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

  getInputMessageTypeTypeSystemName()

  getOutputMessageTypeTypeSystemName()
- The following methods of the ProcessInstanceData object are no longer supported:

  getInputMessageTypeTypeSystemName()

  getOutputMessageTypeTypeSystemName()

- The following methods of the ActivityInstanceData object are no longer supported:

  getInputMessageTypeTypeSystemName()

  getOutputMessageTypeTypeSystemName()
- The following methods of the ActivityServiceTemplateData object are no longer supported:

  getInputMessageTypeTypeSystemName()

Recommended migration action:

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

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

- The following methods are deprecated on the HumanTaskManager bean, and their replacements for use in WebSphere Process Server version 6.0 are given:
  - createMessage(TKIID tkiid, String messageTypeName)

    Use the specific methods createInputMessage(TKIID tkiid), createOutputMessage(TKIID tkiid), createFaultMessage(TKIID tkiid) instead.
  - createMessage(String tkiid, String messageTypeName)

    Use the specific methods createInputMessage(String tkiid), createOutputMessage(String tkiid), createFaultMessage(String tkiid) instead.
- For the Task object, the following methods are no longer supported:

  getInputMessageTypeName()

  getOutputMessageTypeName()

Recommended migration action:

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

The following database views are deprecated:

- DESCRIPTION
- CUSTOM_PROPERTY

Recommended migration action:

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

Programming Model of Java Code Snippets:

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

- The WebSphere Business Integration Server Foundation version 5.1 methods to access custom properties within Java snippet activities are not supported.
- The following getPartnerLink methods are not supported. Because they do not consider partner links declared on the scope level, they can only be used to access partner links declared at the process level.

  EndpointReference getPartnerLink();

  EndpointReference getPartnerLink (int role);

  void setPartnerLink (EndpointReference epr);

Recommended migration action:

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

Application services features

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

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

Recommended migration action:

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

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 InterChange Server version 4.2.0 or later. For more information see "Migrating from WebSphere InterChange Server or WebSphere Business Integration Server Express."
- WebSphere Business Integration Server Foundation versions 5.1 and 5.1.1. For more information see "Migrating from WebSphere Studio Application Developer Integration Edition" on page 187.
- WebSphere MQ Workflow version 3.6. For more information, see "Migrating from WebSphere MQ Workflow" on page 187.

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 InterChange Server to WebSphere Process Server) , the migration steps include using migration tools to convert source artifacts to the new WebSphere Process Server version of the artifacts.

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

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

Migrating from WebSphere InterChange Server or WebSphere Business Integration Server Express

Use the WebSphere Integration Developer wizard or the WebSphere Process Server reposMigrate command to migrate from WebSphere InterChange Server version 4.3 or later or WebSphere Business Integration Server Express version 4.4 or later to WebSphere Process Server 6.2.

About this task

For this version of WebSphere InterChange Server or WebSphere Business Integration Server Express...	Do this
WebSphere InterChange Server version 4.3 or later or WebSphere Business Integration Server Express version 4.4 or later	Use the Migration wizard from WebSphere Integration Developer to migrate WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts into WebSphere Process Server deployable artifacts and place them into projects in the active WebSphere Integration Developer workspace. Alternatively, you can use the reposMigrate command to migrate WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts into WebSphere Process Server deployable artifacts, and optionally deploy them directly to WebSphere Process Server.
WebSphere InterChange Server versions earlier than 4.3 or WebSphere Business Integration Server Express versions earlier than 4.4	First migrate to WebSphere InterChange Server version 4.3 or later or WebSphere Business Integration Server Express version 4.4 or later, and then migrate to WebSphere Process Server.

Related information

Migrating WebSphere InterChange Server using the Migration wizard

WebSphere Integration Developer information center

Premigration considerations

Consider these guidelines for the development of integration artifacts for WebSphere InterChange Server or WebSphere Business Integration Server Express to ease the migration of WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts to WebSphere Process Server.

These recommendations are meant to be used only as a guide. There may be cases where it is necessary to deviate from these guidelines. In these cases care should be taken to limit the scope of the deviation to minimize the amount of rework required to migrate the artifacts. Note that the guidelines outlined here are not all general recommendations for the development of WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts. They are instead limited in scope to those considerations which may affect the ease in which artifacts can be migrated at a future time.

Related concepts

"Troubleshooting migration from WebSphere InterChange Server or WebSphere Business Integration Server Express" on page 179
Find solutions to problems you encounter with migration as well as instructions for turning on logging and tracing.

Related reference

"Postmigration considerations" on page 141
When applications have been migrated from WebSphere InterChange Server or WebSphere Business Integration Server Express to WebSphere Process Server, special attention is required in some areas to enable migrated applications to function in WebSphere Process Server consistently with their intended function due to differences between the architectures of WebSphere Process Server and WebSphere InterChange Server or WebSphere Business Integration Server Express.

Premigration considerations: Access framework clients

Do not develop any new clients adopting the CORBA IDL interface APIs. This is not supported in WebSphere Process Server.

Premigration considerations: Business objects

For the development of business objects, use only the tooling provided to configure artifacts, use explicit data types and lengths for data attributes, and use only the documented APIs.

Business objects within WebSphere Process Server are based on Service Data Objects (SDOs). SDOs use data attributes that are strongly typed. For business objects in WebSphere InterChange Server or WebSphere Business Integration Server Express and adapters, data attributes are not strongly typed, and users sometimes specify string data types for non-string data attributes. To avoid issues in WebSphere Process Server, explicitly specify data types.

Because business objects within WebSphere Process Server might be serialized at runtime as they are passed between components, it is important to be explicit with the required lengths for data attributes to minimize the use of system resources. For this reason, do not use the maximum 255 character length for a string attribute, for example. Also, do not specify zero length attributes which currently default to 255 characters. Instead, specify the exact length required for attributes.

XSD NCName rules apply to business object attribute names in WebSphere Process Server. Therefore, do not use any spaces or ":" in names for business object attributes. Business object attribute names with spaces or ":" are invalid in WebSphere Process Server. Rename business object attributes before migration.

If using an array in a business object, you cannot rely on the order of the array when indexing into the array in Maps and/or Relationships. The construct that this migrates into in WebSphere Process Server does not guarantee index order, particularly when entries are deleted.

It is important to use only the Business Object Designer or the Business Object Designer Express® tool to edit business object definitions, and to use only the published APIs for business objects within integration artifacts.

Premigration considerations: Collaboration templates

When developing WebSphere InterChange Server or WebSphere Business Integration Server Express collaboration templates, follow these guidelines to ensure the best chance of a smooth migration to WebSphere Process Server.

To ensure processes are described appropriately with metadata, always use the Process Designer tool for the creation and modification of collaboration templates, and avoid editing the metadata files directly. Use the Activity Editor tool wherever possible to maximize the use of metadata to describe the required logic.

To minimize the amount of manual rework that may be required in migration, use only the documented APIs within collaboration templates. Avoid the use of static variables. Instead, use non-static variables and collaboration properties to address the requirements of the business logic. Avoid the use of Java qualifiers final, transient and native in Java snippets. These cannot be enforced in the BPEL Java snippets that are the result of migrating the Collaboration Templates.

To maximize future portability, avoid using explicit connection release calls and explicit transaction bracketing (that is, explicit commits and explicit rollbacks) for User Defined Database Connection Pools. Instead, make use of the container-managed implicit connection clean-up and implicit transaction bracketing. Also, avoid keeping system connections and transactions active across Java snippet nodes within a collaboration template. This applies to any connection to an external system, as well as user-defined database connection pools. Operations with an external EIS should be managed within an adapter, and code related to database operation should be contained within one code snippet. This may be necessary within a collaboration which, when rendered as a BPEL business process component may be selectively deployed as an interruptible flow. In this case, the process may be comprised of several separate transactions, with only state and global variable information passed between the activities. The context for any system connection or related transaction which spanned these process transactions would be lost.

Name collaboration template property names in accordance with W3C XML NCName naming conventions. WebSphere Process Server accepts names conforming to those conventions. Any disallowed characters are invalid in BPEL property names that they will be migrated into. Rename properties to remove any disallowed characters before migrating to avoid syntactical errors in the BPEL generated by migration.

Do not reference variables using "this." For example, Instead of "this.inputBusObj" use just "inputBusObj".

Use class-level scoping on variables instead of scenario-scoped variables. Scenario-scoping is not carried forward during migration.

Initialize all variables declared in Java snippets with a default value: "Object myObject = null;" for example. Be sure all variables are initialized during declaration before migrating.

Ensure that there are no Java import statements in the user modifiable sections of your collaboration templates. In the definition of the collaboration template, use the import fields to specify Java packages to import.

Do not set incoming business object values to be stored in the *triggeringBusObj* variable. Within WebSphere InterChange Server or WebSphere Business Integration Server Express, the *triggeringBusObj* is read-only and its values cannot be overwritten, so any incoming business object values will not be saved. If the *triggeringBusObj* is used as the receiving variable for an incoming business object on an inbound service call, then after migration the behavior of the inbound

service call will be different: within the BPEL process, the incoming value from the inbound service call will overwrite the value stored in *triggeringBusObj*.

Premigration considerations: Common code utilities

IBM recommends that you avoid the development of common code utility libraries for use across integration artifacts within the WebSphere InterChange Server or WebSphere Business Integration Server Express environment. Consider using EJBs running in WebSphere Application Server to encapsulate the logic, and use web service calls to invoke them from WebSphere InterChange Server or WebSphere Business Integration Server Express.

While it is possible that some common code utility libraries may run appropriately on WebSphere Process Server, you will be responsible for the migration of the custom utilities.

Premigration considerations: Database connection pools

A WebSphere InterChange Server or WebSphere Business Integration Server Express database connection pool within a map or collaboration template will be rendered as a standard JDBC resource in WebSphere Process Server. However, the way connections and transactions are managed might differ between WebSphere InterChange Server or WebSphere Business Integration Server Express and WebSphere Process Server, so you should avoid keeping database transactions active across Java snippets.

User-defined database connection pools are useful within maps and collaboration templates for simple data lookups and for more sophisticated state management across process instances. A database connection pool in WebSphere InterChange Server or WebSphere Business Integration Server Express will be rendered as a standard JDBC resource in WebSphere Process Server, and the basic function will be the same. However, the way connections and transactions are managed may differ.

To maximize future portability, avoid keeping database transactions active across Java snippet nodes within a collaboration template or map. For example, code related to obtaining a connection, beginning and ending a transaction, and releasing the connection should be in one code snippet.

Premigration considerations: General development

Follow these recommended practices when developing WebSphere InterChange Server or WebSphere Business Integration Server Express modules to ease future migration to WebSphere Process Server.

Several considerations apply broadly to developing most of the integration artifacts. In general, artifacts that leverage the facilities provided by WebSphere InterChange Server or WebSphere Business Integration Server Express tooling and conform to the metadata models enforced by the tooling will migrate most smoothly. Also, artifacts with significant extensions and external dependencies are likely to require more manual intervention when migrating.

In general, IBM recommends that you do the following:

- Document the system and component design
- Use the development tooling to edit integration artifacts
- Leverage recommended practices for defining rules with the tooling and Java snippets

It is important for integration solutions to adhere to the programming model and architecture provided by WebSphere InterChange Server or WebSphere Business Integration Server Express. Each of the integration components within WebSphere InterChange Server or WebSphere Business Integration Server Express plays a well-defined role within the architecture. Significant deviations from this model will make it more challenging to migrate content to the appropriate artifacts on WebSphere Process Server.

Another general practice which will improve the success of future migration projects is to document the system design. Be sure to capture the integration architecture and design, including functional design and quality of service requirements, the interdependencies of artifacts shared across projects, and also the design decisions that were made during the deployment. This will assist in system analysis during migration, and will minimize any rework efforts.

For creating, configuring, and modifying artifact definitions, use only the development tooling provided. Avoid manual manipulation of artifact metadata (for example, editing XML files directly), which may corrupt the artifact for migration.

IBM suggests the following when you are developing Java code within collaboration templates, maps, common code utilities, and other components:

- Use only the published APIs.
- Use Activity Editor.
- Use adapters to access EISs.
- Avoid external dependencies in Java snippet code.
- Adhere to Java EE develop practices for portability.
- Do not spawn threads or use thread synchronization primitives. If you must, these will need to be converted to use Asynchronous Beans when you migrate.
- Do not do any disk I/O using java.io.* Use JDBC to store any data.
- Do not perform any functions that may be reserved for an EJB container such as socket I/O, classloading, loading native libraries, and so forth. If you must, these snippets would need manual conversion to use EJB container functions when you migrate.

Use only the APIs published in the WebSphere InterChange Server or WebSphere Business Integration Server Express product documentation for the artifacts. These are outlined in detail in the WebSphere InterChange Server or WebSphere Business Integration Server Express development guides. Compatibility APIs will be provided in WebSphere Process Server for published WebSphere InterChange Server or WebSphere Business Integration Server Express APIs. Although WebSphere InterChange Server and WebSphere Business Integration Server Express have many internal interfaces which you might wish to use, IBM discourages this practice because these interfaces are not guaranteed to be supported in the future.

When designing business logic and transformation rules in maps and collaboration templates, try to avoid field developed common code utility libraries, included as a Java archive (*.jar) file in the class path of WebSphere InterChange Server or WebSphere Business Integration Server Express, as these will need to be migrated manually.

Use the Activity Editor tool to the greatest extent possible. This will ensure that the logic is described through metadata which can more readily be converted to the new artifacts.

In any Java code snippets that may need to be developed, IBM recommends that the code be as simple and atomic as possible. The level of sophistication in the Java code should be on the order of scripting, involving basic evaluations, operations, and computations, data formatting, type conversions, and so forth. If more extensive or sophisticated application logic is required, consider using EJBs running in WebSphere Application Server to encapsulate the logic, and use web service calls to invoke it from WebSphere InterChange Server or WebSphere Business Integration Server Express. Use standard JDK libraries rather than third party or external libraries which would need to be migrated separately. Also, collect all related logic within a single code snippet, and avoid using logic where connection and transaction contexts span multiple code snippets. With database operations, for example, code related to obtaining a connection, beginning and ending a transaction, and releasing the connection should be in one code snippet.

In general, ensure that code which is designed to interface with an Enterprise Information System (EIS) is placed within adapters, and not within maps or collaboration templates. This is generally a recommended practice for architecture design. Also, this will help avoid prerequisites for third party libraries and related considerations within the code, such as connection management and possible Java Native Interface (JNI) implementations.

Make the code as safe as possible by using appropriate exception handling. Also make the code compatible to run within a Java EE application server environment, even though it is currently running within a Java SE environment. Adhere to Java EE development practices, such as avoiding static variables, spawning threads, and disk I/O. While these are generally good practices to adhere to, they can improve portability.

Premigration considerations: Maps

When developing WebSphere InterChange Server or WebSphere Business Integration Server Express maps, follow these guidelines to ensure the best chance of a smooth migration to WebSphere Process Server.

To ensure maps are described appropriately with metadata, always use the Map Designer or the Map Designer Express tool for the creation and modification of maps, and avoid editing the metadata files directly. Use the Activity Editor tool wherever possible to maximize the use of metadata to describe the required logic.

When referencing child business objects in a map, use a submap for the child business objects.

Avoid using Java code as the "value" in a SET since that is not valid in WebSphere Process Server. Use constants instead. For example, if the set value is "xml version=" + "1.0" + " encoding=" + "UTF-8" this will not validate in WebSphere Process Server. Instead, change it to "xml version=1.0 encoding=UTF-8" before you migrate.

To minimize the amount of manual rework that may be required in migration, use only the documented APIs within maps. Avoid the use of static variables. Instead, use non-static variables. Avoid the use of Java qualifiers final, transient and native in map custom code.

If using an array in a business object, do not rely on the order of the array when indexing into the array in maps. The construct that this migrates into in WebSphere Process Server does not guarantee index order, particularly when entries are deleted.

To maximize future portability, avoid using explicit connection release calls and explicit transaction bracketing (that is, explicit commits and explicit rollbacks) for User Defined Database Connection Pools. Instead, make use of the container-managed implicit connection clean-up and implicit transaction bracketing. Also, avoid keeping system connections and transactions active in custom map steps across transformation node boundaries. This applies to any connection to an external system, as well as user-defined database connection pools. Operations with an external EIS should be managed within an adapter, and code related to database operation should be contained within one custom step.

Do not use inner classes in your maps. The migration command (reposMigrate) does not migrate inner classes and you will receive errors if your maps contain them. In a WebSphere InterChange Server or WebSphere Business Integration Server Express repository, an inner class could be defined in a node and referenced by other nodes within the same collaboration template. In WebSphere Process Server, an inner class defined in a BPEL component cannot be used by other components. Due to this limitation, inner classes are not translated and must be dealt with manually. Recommended changes include packaging the inner class code in a library as an external class, or removing the inner class declaration, resolving any errors, and placing the code as needed throughout the BPEL.

Premigration considerations: Reverse maps on connectors

Determine whether applications with response flows have reverse maps associated with their outbound connectors. If they do not, then a generic SMO is returned in the response.

Some of the applications you have created for WebSphere InterChange Server or WebSphere Business Integration Server Express might not include reverse maps on outbound connector ports. This situation would occur where you are not concerned with the contents of the results returned by the response. Be aware, however, that WebSphere Process Server requires that a valid SMO be returned for all mediation flow components with two-way invocations. Therefore, WebSphere Process Server returns a generic SMO in mediation flow components that do not have reverse maps.

Premigration considerations: Preventing database collisions

Prevent database collisions from occurring by scheduling events to occur at least two seconds apart.

If your migrated applications cause multiple events to occur at the same time to WebSphere Business Integration components, this could cause database collisions, or deadlocks. These occur when the WebSphere Process Server Application Scheduler (AppScheduler) schedules multiple events to occur at exactly the same time. When a deadlock occurs, the event that caused it is rolled back and attempted again as soon as possible. This cycle continues until each of the threads attempting to access the database successfully updates it.

For example:

```
AppScheduler E com.ibm.wbiserver.scheduler.AppSchedulerMB process CWLWS0021E:
The AppSchedulerMB.process method has generated an exception.
WSRdbXaResour E DSRA0304E: XAException occurred. XAException contents and
  details are:
The DB2 Error message is : Error executing a XAResource.end(), Server returned
XA_RBDEADLOCK The DB2 Error code is : -4203
The DB2 SQLState is : null
```

To prevent this from occurring, schedule the events to occur far enough apart so that deadlocks do not occur. IBM recommends that you schedule events to occur at least two seconds apart; however, the amount of time you need will vary depending on other factors in your environment that affect performance such as database size, hardware, connection speed and other factors.

Premigration considerations: Relationships

While relationship definitions can be migrated for use in WebSphere Process Server, the relationship table schema and instance data may be reused by WebSphere Process Server, and shared concurrently between WebSphere InterChange Server or WebSphere Business Integration Server Express and WebSphere Process Server.

For relationships, use only the tooling provided to configure the related components, and use only the published APIs for relationships within integration artifacts.

Use only the Relationship Designer or the Relationship Designer Express tool to edit relationship definitions. In addition, allow only WebSphere InterChange Server or WebSphere Business Integration Server Express to configure the relationship schema, which is generated automatically upon deployment of relationship definitions. Do not alter the relationship table schema directly with database tools or SQL scripts.

If you must manually modify relationship instance data within the relationship table schema, be sure to use the facilities provided by the Relationship Manager.

Use only the published APIs for relationships within integration artifacts.

Migrating WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts with the reposMigrate command

Migrate WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts to WebSphere Process Server artifacts with the **reposMigrate** command.

Before you begin

Note: The functionality of the reposMigrate command is also available from WebSphere Integration Developer with a supporting wizard (graphical user interface). See the WebSphere Integration Developer information center for more information.

The **reposMigrate** command requires as input a WebSphere InterChange Server or WebSphere Business Integration Server Express repository JAR file. This JAR file should be self-contained with respect to the applications being migrated. That is, all artifacts referenced by any of the artifacts in the JAR file must also be contained in the JAR file.

To ensure that the repository JAR file that will be generated is self-contained, run the **repos_copy** command with the `-vr` option before exporting the server repository. This validates the repository. If the repository is valid then repos_copy writes the following output to the console: `Validation Succeeded. All`

`Dependencies Resolved.` If the repository is not valid then **repos_copy** prints a list of the dependencies that must be resolved. Resolve the dependencies prior to exporting the repository.

Export the repository artifacts and create the repository JAR file, using the WebSphere InterChange Server or WebSphere Business Integration Server Express **repos_copy** command with the -o option (See the WebSphere InterChange Server or WebSphere Business Integration Server Express v4.3 documentation for more details, including how to export individual components).

About this task

The **reposMigrate** command will convert the WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts in a JAR file into WebSphere Process Server deployable artifacts. These artifacts are modules created as one or more JAR files. A JAR file is created for each collaboration object and for each connector definition that has been migrated. For other artifacts such as business objects, maps and relationships, a copy of all of these artifacts generated from the input JAR file will be included in each JAR file generated. If no collaboration objects or connectors are migrated, a single JAR file is created containing a module of all the shared artifacts. After the new JAR files are created, you will use the **serviceDeploy** command to generate the EAR files that can be deployed in WebSphere Process Server.

For WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts that have no corresponding artifact in WebSphere Process Server, a Jython script is generated during migration that can be run using the **wsadmin** command to create WebSphere Process Server configuration definitions corresponding to the original WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts.

Procedure

1. Identify the JAR file containing the pre-exported WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts that are to be converted to WebSphere Process Server deployable artifacts.
2. Invoke the **reposMigrate** command from a command-line prompt. Type the command at a command prompt in WebSphere Process Server, with the required arguments and any optional arguments you require. Refer to reposMigrate command for more information.
3. If desired, edit the resulting JAR file.
4. Run serviceDeploy to create a deployable EAR file for each JAR file.

   Note: The support in the WebSphere Process Server Runtime to handle migrated WebSphere InterChange Server applications relies on the default naming convention used by the serviceDeploy command. IBM recommends that you do not specify the **serviceDeploy -outputApplication** parameter when building migrated projects with the **serviceDeploy** command, so that it will generate its default output filenames.

   For more information, refer to the WebSphere Process Server serviceDeploy command in the *Reference* PDF file.
5. Use the administrative console or the **wsadmin** command to install the EAR files on WebSphere Process Server. Use the **wsadmin** command to execute the

InstallAdministrativeObjects.py script. This will create resources in the WebSphere Process Server system for all target resources such as JDBC data sources and WBIScheduler entries.

Example

You can use the **reposMigrate** command to migrate existing WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts directly to a running WebSphere Process Server:

1. Open a command prompt in WebSphere Process Server.
2. Issue the **reposMigrate** command with the following mandatory parameters:
   install_root\bin\reposMigrate *SourceArtifactJAR OutputArtifactDirectory*

The reposMigrate command builds the generated artifacts as follows:

- For each WebSphere InterChange Server or WebSphere Business Integration Server Express collaboration object and connector definition in the input JAR file, **reposMigrate** creates a JAR file from the migrated artifacts.
- For other artifacts such as business objects, maps and relationships, a copy of all of these artifacts generated from the input JAR file will be included in each JAR file generated. If no collaboration objects nor connector definitions were in the input, a single JAR file will be created with all the shared artifacts.

What to do next

The default behavior of the **reposMigrate** command is to log errors for the migration of each individual artifact and continue to migrate the remainder of the artifacts. You should check output messages for errors after the execution completes. To override this default behavior and force **reposMigrate** to end processing when the first artifact that cannot be migrated is encountered, specify the **-fh** (halt at first failure) flag. You can run **reposMigrate** from the beginning to retry after a failed execution.

Related reference

"Postmigration considerations"
When applications have been migrated from WebSphere InterChange Server or WebSphere Business Integration Server Express to WebSphere Process Server, special attention is required in some areas to enable migrated applications to function in WebSphere Process Server consistently with their intended function due to differences between the architectures of WebSphere Process Server and WebSphere InterChange Server or WebSphere Business Integration Server Express.

Related information

Wsadmin tool

reposMigrate command

WebSphere InterChange Server v4.3 documentation

WebSphere Integration Developer information center

Postmigration considerations

When applications have been migrated from WebSphere InterChange Server or WebSphere Business Integration Server Express to WebSphere Process Server, special attention is required in some areas to enable migrated applications to function in WebSphere Process Server consistently with their intended function due

to differences between the architectures of WebSphere Process Server and WebSphere InterChange Server or WebSphere Business Integration Server Express.

You should be aware of the information described in the following sections if it applies to your application and environment:

Security

Additional security configuration is required for your applications to have the same security levels set as they had when running in WebSphere InterChange Server or WebSphere Business Integration Server Express. For details on this configuration, refer to "Configuring global security after WebSphere InterChange Server or WebSphere Business Integration Server Express migration" on page 153.

Handling existing database connections, relationships, and scheduled events (InstallAdministrativeObjects.py script)

The Jython script InstallAdministrativeObjects.py is generated during migration. This script has three purposes: It allows migration of WebSphere InterChange Server or WebSphere Business Integration Server Express scheduler entries that have no corresponding artifact in WebSphere Process Server; it allows the use of existing DBConnection pools; and it allows the use of an existing relationship database. You can run the script with the wsadmin command to create WebSphere Process Server configuration definitions corresponding to the original WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts. A copy of InstallAdministrativeObjects.py is included wherever the shared artifacts are included. That is, the script is included in every JAR file created by the reposMigrate command, and it is put in the shared library project specified during import in WebSphere Integration Developer. An InstallAdministrativeObjects.py script is always generated even if there are no artifacts that require it. This script can be modified to add or delete entries before using the wsadmin command to execute it.

For more information about using the wsadmin command, see wsadmin tool .

Handling existing WebSphere InterChange Server or WebSphere Business Integration Server Express database connection pools

To preserve existing WebSphere InterChange Server or WebSphere Business Integration Server Express DataBase Connection pools for use by WebSphere Process Server you can run the InstallAdministrativeObjects.py script with the wsadmin command to create the connection pools in WebSphere Process Server. If an appropriate JDBC provider is not defined, this script will use default JDBC provider templates to create JDBC providers. A side effect of using these default templates is that WebSphere Process Server creates an empty, sample data source definition. This sample data source is not used; you must delete it to prevent exceptions from occurring during server start because it does not specify all of the information required for a data source.

In the WebSphere InterChange Server or WebSphere Business Integration Server Express environment, resources are defined only once for the entire system. To simulate this in the WebSphere Process Server environment, the InstallAdministrativeObjects.py script defines resources at the cell scope. WebSphere variables are predefined at the node scope in the WebSphere Process Server system for use by JDBC providers created from the default JDBC provider templates. These variables are defined at the node scope so that they can be customized for each node. Because of this scoping discrepancy, you will need to do one of the following:

- Define the WebSphere Variables needed by the created JDBC providers at the cell scope.
- Run the InstallAdministrativeObjects.py script and then move the JDBC providers to the node scope.

Use the administrative console to examine the JDBC providers that are generated to determine which WebSphere variables are needed. From the administrative console, select **Environment > WebSphere Variables** to create any required variables. For more information, see Defining WebSphere variables in the WebSphere Application Server Network Deployment, version 6.1 information center.

Here is an example of what the generated InstallAdministrativeObjects.py script might contain to generate the JDBC connector pool:

```
dsName = "sqls"
create_datasource(dsName, JNDI_PREFIX + dsName, DATASOURCE_DESCRIPTION,
MS_SQL_JDBC_PROVIDER_NAME, MS_SQL_JDBC_PROVIDER_TYPE, "icsadmin",  "icsadmin",
4, 50, "qaxs17", "1433", "wicsrepos")
```

For more information about the wsadmin command, see wsadmin tool .

Using an existing WebSphere InterChange Server or WebSphere Business Integration Server Express relationship database

To use an existing WebSphere InterChange Server or WebSphere Business Integration Server Express relationship database in WebSphere Process Server, you can use the InstallAdministrativeObjects.py script with the wsadmin command to create the data source and relationship configuration information in WebSphere Process Server. Normally, WebSphere Process Server automatically creates the configuration information for the migrated relationships when they are deployed. To be able to use the existing database, the InstallAdministrativeObjects.py script has to create the database connection for the existing WebSphere InterChange Server or WebSphere Business Integration Server Express relationship database and the relationship configuration information in WebSphere Process Server. Runs the InstallAdministrativeObjects.py script before you deploy the migrated components. Then, when WebSphere Process Server deploys the relationships it will use the configuration information that was generated by the script.

Here is an example of what the generated InstallAdministrativeObjects.py script might contain to generate the relationship database connection:

```
dsName = "ContactR"
create_datasource(dsName, JNDI_PREFIX + dsName, DATASOURCE_DESCRIPTION,
MS_SQL_JDBC_PROVIDER_NAME, MS_SQL_JDBC_PROVIDER_TYPE, "icsadmin",  "icsadmin",
-1, -1, "9.26.230.56", "1433", "wicsrepos")

create_relationship("ContactR", "jdbc/wbi60migration/ContactR", "false")
create_role("ContactR", "ID1", "", "null", "", "null")
create_attribute("ContactR", "ID1", "JtextEmployeeID")
create_role("ContactR", "ID2", "", "null", "", "null")
create_attribute("ContactR", "ID2", "EmployeeID")
create_role("ContactR", "ID3", "", "null", "", "null")
create_attribute("ContactR", "ID3", "EmployeeID")
```

For more information about the wsadmin command, see wsadmin tool .

Migrating scheduled events

Because there is no WebSphere Process Server component that corresponds to WebSphere InterChange Server or WebSphere Business Integration Server Express scheduler entries, migration of WebSphere InterChange Server or WebSphere Business Integration Server Express scheduler entries is accomplished by extracting the pertinent data from the existing WebSphere InterChange Server or WebSphere Business Integration Server Express repository JAR file and creating corresponding entries in the WebSphere Process Server scheduler tables in the WebSphere Process Server Common database. The data is represented in string form in the Jython script. To create the scheduler entries in the WebSphere Process Server database, you can run the InstallAdministrativeObjects.py script with the wsadmin command.

Here is an example of what the generated InstallAdministrativeObjects.py script might contain to generate the scheduler entry:

```
create_scheduler_entry("true", "stop", "JDBCConnector", "Connector",
"2006-09-07T10:44:29.000PDT", "undefined", 0, 0)
   create_scheduler_entry("true", "start", "JTextConnector", "Connector",
"2006-09-07T10:47:06.000PDT", "undefined", 0, 0)
   create_scheduler_entry("true", "stop", "jtext_jdbcCollab", "Collaboration",
"2006-09-07T10:48:10.000PDT", "undefined", 0, 0)
   create_scheduler_entry("true", "start", "jtext_jdbcCollab", "Collaboration",
"2006-09-07T10:48:10.000PDT", "undefined", 0, 0)
   create_scheduler_entry(true, "START", "JDBCConnector", "Connector",
"2006-10-22T12:34.56.789CDT", "MINUTES", 20, 0):
```

Access Enterprise JavaBean (EJB) support

WebSphere InterChange Server or WebSphere Business Integration Server Express supports the triggering of collaborations by client code via the Java EE EJB (Enterprise JavaBeans) protocol. Support for this method of triggering collaborations is referred to as "AccessEJB" or "AccessEJB for EJB" support. For compatibility with earlier versions, WebSphere Process Server provides support for AccessEJB. The AccessEJB support assumes that the SCA BPEL modules to be invoked were generated by the WebSphere InterChange Server or WebSphere Business Integration Server Express migration tools described in this documentation. The mapping from the collaboration name and port name (that is, the input parameters for the AccessEJB) to the SCA module name, interfaces and business object types assume the conventions used by the migration tools. The AccessEJB support in WebSphere Process Server is delivered in the AccessEJB.zip project interchange file. This file is located in the *install_root*/HeritageAPI directory. The AccessEJB support consists of an EJB (AccessEJB) which references an SCA module project (DynamicRouting) that invokes the SCA BPEL module. This SCA BPEL module is the migrated version of the collaboration that was invoked in WebSphere InterChange Server or WebSphere Business Integration Server Express. The DynamicRouting module uses a selector component to select the correct SCA target based on the collaboration name and port name passed to the AccessEJB. To enable AccessEJB support in WebSphere Process Server, do the following:

1. Import the WebSphere InterChange Server or WebSphere Business Integration Server Express repository containing the collaboration that is the target of the AccessEJB invocation into WebSphere Integration Developer.
2. Import the AccessEJB.zip project interchange file into WebSphere Integration Developer.
3. Open the DynamicRouting project and update the selector table to include the migrated module that is to be invoked via the AccessEJB.
4. Go to the migrated project containing the BPEL component to be invoked via the AccessEJB EJB, and drag the export that references the BPEL module over to the DynamicRouting project.
5. Repeat steps 3 and 4 for each BPEL module that is to be accessible via AccessEJB.
6. Build the project and deploy it to the WebSphere Process Server server.
7. Ensure that any required data handlers are provided in the runtime class path of the WebSphere Process Server server.
8. To enable your Access client to use WebSphere Process Server, ensure that it points to the WebSphere Process Server server and uses the JNDI name com/crossworlds/access/business/cwsession/CwSession when looking up the Access EJB.

DynamicSend API configuration

In WebSphere InterChange Server or WebSphere Business Integration Server Express, the DynamicSend API can be used to directly invoke one collaboration from another. The collaboration to be invoked does not have to be predetermined; instead, it can be determined dynamically at runtime. The support for the DynamicSend API in WebSphere Process Server uses the DynamicRouting project described in "“Access Enterprise JavaBean (EJB) support” on page 145." Follow the instructions in “Enabling the BaseCollaboration.dynamicSend method call” to enable the DynamicSend API to be able to invoke the specified BPEL modules.

Enabling the BaseCollaboration.dynamicSend method call

To enable the WebSphere InterChange Server or WebSphere Business Integration Server Express BaseCollaboration.dynamicSend method call to work correctly after migration you must modify the DynamicRouting Projects in the AccessEJB Project Interchange file. This requires two main procedures:

1. Migrating the WebSphere InterChange Server or WebSphere Business Integration Server Express repository.
2. Enabling the DynamicSend API.

To migrate the WebSphere InterChange Server or WebSphere Business Integration Server Express repository:

1. Import the WebSphere InterChange Server or WebSphere Business Integration Server Express repository containing the collaboration that invokes DynamicSend API into WebSphere Integration Developer.
2. Import the WebSphere InterChange Server or WebSphere Business Integration Server Express repository containing the collaboration or connector that is the target of the DynamicSend API invocation into WebSphere Integration Developer.
3. Build all, and correct all errors.

To enable the DynamicSend API:

1. Import the `AccessEJB.zip` project interchange file into WebSphere Integration Developer.
2. Open the DynamicRouting project, and add the WebSphere InterChange Server or WebSphere Business Integration Server Express shared library into the DynamicRouting project's dependencies.
3. Go to the migrated module containing the component to be invoked through the BaseCollaboration.dynamicSend method, and drag the export that references the module over to the DynamicRouting project. Choose **Import with SCA Binding** and then click **OK**.
4. In the DynamicRouting Assembly Diagram window, copy and paste `PreRoute_TargetCollab_TargetPort` and then rename the newly created copy to `PreRoute_`*`ModuleName_ExportName`* (the name of the copied import will be `PreRoute_TargetCollab_TargetPortCopy`).
5. On `PreRoute_`*`ModuleName_ExportName`*, left click on the reference, which is the small box attached to the right containing `1.1`. Right click and choose **Delete**.
6. Wire the `PreRoute_`*`ModuleName_ExportName`* to the import generated in step 3. Respond with "no" to the Java WSDL reference question.
7. Rename the import to *`ModuleName_ExportName`*. Save the changes to the Assembly Diagram.

8. Update the selector table in the DynamicRouting project to include the migrated module that is to be invoked through the DynamicSend API.
   a. Switch to the Java Perspective Package Explorer View. Expand `DynamicRouting/com.ibm` and open `RoutingSelector.selt` with the text editor.
   b. Copy the OperationSelectionRecord block and paste the entire block immediately following the existing block.
   c. In the new block, change `componentName="PreRoute_TargetCollab_TargetPort"` to `componentName="PreRoute_`*`ModuleName_ExportName`*`"`. Also in the new block, change `value="TargetCollab_TargetPort"` to `value="`*`ModuleName_ExportName`*`"`.

```
<OperationSelectionRecord>
      <SelectionKey>
      <SelectionKeyElement xsi:type="selt:StringSingletonKey" value=
           "TargetCollab_TargetPort"/>
      </SelectionKey>
      <SelectionData xsi:type="selt:SCAInternalComponent"
            componentName="PreRoute_TargetCollab_TargetPort"/>
      </OperationSelectionRecord>
      <OperationSelectionRecord>
      <SelectionKey>
      <SelectionKeyElement xsi:type="selt:StringSingletonKey"
             value="ModuleName_ExportName"/>
      </SelectionKey>
      <SelectionData xsi:type="selt:SCAInternalComponent" componentName
             ="PreRoute_ModuleName_ExportName"/>
</OperationSelectionRecord>
```

   d. Save and close `RoutingSelector.selt`.
9. Generate the implementation file.
   a. Expand **com.ibm.sel** and copy `PreRoute_TargetCollab_TargetPortImpl.java` and paste it to the same location. Name the newly created Java file `PreRoute_`*`ModuleName_ExportName`*`Impl.java`.
   b. Edit `PreRoute_`*`ModuleName_ExportName`*`Impl.java`. Change the method name of `locateService.TestBOInterfacePartner` to `locateService_`*`InterfaceName`*`Partner` (*InterfaceName* is the method). Change `TestBOInterfacePartner` to *`InterfaceName`*`Partner`.
   c. Search for "`locateService_TestBOInterfacePartner`" in `PreRoute_`*`ModuleName_ExportName`*`Impl.java`, and change its name to `locateService_`*`InterfaceName`*`Partner`.
   d. Search for "`this.locateService_InterfaceNamePartner().invoke("Sync", tmpres)`" in *`PreRoute_ModuleName_ExportName`*`Impl.java`, and change its name to "`this.locateService_InterfaceNamePartner.invoke("Sync_ExportName", tmpres)`", then save it.
10. Switch back to the Business Integration Perspective. Open the DynamicRouting Assembly Diagram. Click on **PreRoute*_ModuleName_ExportName***. Open **Properties** and select **Implementation**. In the **Class:** field, enter `com.ibm.sel.PreRoute_`*`ModuleName_ExportName`*`Impl`.
11. Save all changes.
12. Repeat steps 3 to 11 for any other modules that you want to call from the BaseCollaboration.dynamicSend Method. There is currently no way to

"dynamically look these modules up" if you do not add them to the DynamicRouting Table so you can access them during run time.

13. For the project that calls the dynamicSend API, do the following
    a. Copy and paste Interface "RoutingPacket" from Module DynamicRouting.
    b. In the component that calls the dynamicSend method. Add the newly copied interface "RoutingPacket" to Reference_Partners, and rename it "RoutingPacketPartner."
    c. Save it.
    d. Open the Assembly Diagram. Drag "RoutingInput" from DynamicRouting. Choose "Import with SCA Binding" and click "OK". Rename it from "Import1" to "DynamicRouting".
    e. Delete and re-drag the component that calls the dynamicSend API to the Assembly Diagram window, wire the Reference "RoutingPacketPartner" to "DynamicRouting" and re-wire the other references.
14. Save all and build, then correct all errors. Export all the modules to EAR files.

Event sequencing migration

Methods are available for sequencing events with WebSphere Process Server in ways similar to the way you could with WebSphere InterChange Server or WebSphere Business Integration Server Express. Articles on this subject that you might find helpful are available from the IBM developerWorks web site. Search in the "Technical Library" at http://www.ibm.com/developerworks.

Failed events

Methods for handling failed events in WebSphere Process Server are described in article(s) that you might find helpful on the IBM developerWorks web site. Search in the "Technical Library" at http://www.ibm.com/developerworks.

Map migration

WebSphere InterChange Server or WebSphere Business Integration Server Express migration converts WebSphere InterChange Server or WebSphere Business Integration Server Express maps into WebSphere Process Server maps. Two output maps are generated: the business graph map and the business object map. The business graph map calls the business object map as a submap. All the business graph maps are identical in structure. Differences include names, names of the submap they call, or ASI information on the verb attribute. These business graph maps are present only to satisfy the necessary mapping steps that can only be done at the business graph level. The business object maps are each unique and are the migrated form of the WebSphere InterChange Server or WebSphere Business Integration Server Express map. If the WebSphere InterChange Server or WebSphere Business Integration Server Express input map contains custom messages for the supported WebSphere InterChange Server or WebSphere Business Integration Server Express API log methods, these messages will be converted into a properties file.

Collaboration migration

Collaboration Templates: The WebSphere InterChange Server or WebSphere Business Integration Server Express to WebSphere Process Server migration tools migrate WebSphere InterChange Server or WebSphere Business Integration Server Express Collaboration templates into WebSphere Process Server BPEL files. One

BPEL file is created for each triggering port defined in a collaboration template, and its name is based on the following naming convention: *CollaborationTemplateName_TriggeringPortName*. Each BPEL file receives a business object type that is based on the business object type associated with the triggering port. For example, if the triggering port takes a business object type of Customer, then the BPEL file that is created will have a "TriggeringBusObj" variable type of Customer.

Collaboration Objects: The WebSphere InterChange Server or WebSphere Business Integration Server Express to WebSphere Process Server migration tools migrate collaboration objects into several service component architecture (SCA) components. Currently, migration supports collaboration objects that reference collaboration templates as follows:

- Supported:
  - One or more triggering ports, no correlation sets and no asynchronous in calls
  - Exactly one triggering port, correlation sets and asynchronous in calls
- Unsupported:
  - Migration does not support the case of one or more triggering ports, correlation sets and asynchronous in calls. In this case, the resulting artifacts are migrated as if they were the first case listed above. Additionally you will need to manually create the missing SCA components and wire them together appropriately.

SCA Components:

- Exports: An export is created for every triggering port defined in the collaboration template associated with the collaboration object. The export name is *TriggeringPortName*.
- Export to BPEL: An interface map is generated that maps the data from the export to the BPEL file. The interface map name is *Export_To_BPELname.* When there is exactly one triggering port and the collaboration template has an asynchronous in call, additional SCA components are created. Instead of having just one interface map, the migration results in two interface maps: one for synchronous calls, and another for asynchronous calls. A Java component is used to decide which of these two interface maps to follow.
- BPEL: For every triggering port, the export will be wired to an interface map and that interface map will map to an instance of the BPEL file.
- BPEL to import: Every port, triggering and non-triggering, has an interface map mapping the BPEL file to the import. The interface map name is *BPEL_to_Port*.
- Import: Finally, an import file is created. The import name is *ConnectorName_BONameBG*.

For further detailed information on how collaboration templates are migrated to WebSphere Process Server BPEL files, see the IBM developerWorks article Migrating WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts to WebSphere Process Server artifacts, Part 1: Migrating collaboration templates to BPEL .

BPEL variables must be defined after migration

Problem: A variable that is not defined in the Ports definitions of the WebSphere InterChange Server or WebSphere Business Integration Server Express Collaboration Template is used to invoke a partner. After migration, the variable is referenced in the business process execution language (BPEL) invoke but has not

been set up as a BPEL variable, so it is flagged as an error when executing the serviceDeploy command against the module or after building the module in WebSphere Integration Developer. **Cause:** When invoking a partner from a BPEL process in WebSphere Process Server, any object used in the invoke must be declared as a BPEL variable so that the type of the object being used can be determined. During migration, only the Ports declarations in the Collaboration Template are examined to determine what BPEL variables need to be declared. For global variables, or variables declared in snippets elsewhere in the ICS Collaboration Template definition, the migration code cannot reliably determine the object type, so BPEL variables are not declared for these in the BPEL file generated by migration. **Solution:** After migration, you must define the variable as a BPEL variable for the variable to be referenced during an invoke.

Enabling logError API e-mail notification on WebSphere Process Server

Problem: After migration to WebSphere Process Server, the WebSphere InterChange Server or WebSphere Business Integration Server Express logError API does not send an e-mail to a list of users that has been configured in WebSphere InterChange Server or WebSphere Business Integration Server Express. **Cause:** In WebSphere InterChange Server or WebSphere Business Integration Server Express, you could configure the API call logError to send an error e-mail to a specified list of users. However, this list of users, configured on the server, is not accessible to the migration code so it must be set up manually in WebSphere Process Server. **Solution:**To enable the WebSphere InterChange Server or WebSphere Business Integration Server Express logError e-mail notification functionality in WebSphere Process Server, a new BPEL environment variable called *LOGERROR_EMAIL_LIST* is created in each BPEL file generated by migration. Set this variable with the list of e-mail users needing to receive log error e-mails. Separate the names within the list with a comma.

Handling asynchronous calls in WebSphere Process Server

Problem: Async-in events act as triggering events when both types of events can be received on the same connector. **Cause:** If both async-in and triggering events can be received on the same connector, the migrated application cannot determine which events are which type. By default, all events are treated as triggering events in a migrated application in this scenario. **Solution:** Application-specific logic that can determine if an event is async-in or triggering must be added to the migrated application. Migrated modules that can receive triggering events and async-in events on the same connector will have a component named JavaSelector. The implementation code for the JavaSelector component will contain the AsyncIn() method shown below. This method must be updated with logic to check if events are async-in or triggering. This logic will be specific to each application and will be based on the nature of the events being handled.

```
/** * Method generated to support async inbound service call routing */
public boolean isAsyncIn()
{ //Add custom code here
   //TODO
   return false;
}
```

Enabling AppScheduler to start after network deployment upgrade

Problem: After migrating a WebSphere Process Server 6.0.1.x network deployment configuration to WebSphere Process Server 6.1, the AppScheduler fails to start on

WebSphere Process Server 6.0.1.x servers and clusters that have not been upgraded. An Exception similar to the following will be generated:

```
WSVR0040E: addEjbModule failed for WBISchedulerEJB.jar
   [class com.ibm.ws.runtime.component.
DeployedEJBModuleImpl] java.lang.NoClassDefFoundError:
   com/ibm/wbiserver/scheduler/common/AppSchedulerException
```

Cause: After migrating the WebSphere Process Server 6.0.1.x network deployment configuration to WebSphere Process Server 6.1, the AppScheduler application looks for the AppSchedulerException Class in the WebSphere Process Server 6.0.1.x version of the `wbischedulercommon.jar` file and fails to find it in the local system's *install_root*/lib directory. It then throws a `java.lang.NoClassDefFoundError: com/ibm/wbiserver/scheduler/common/AppSchedulerException` exception. **Solution:** Replace the WebSphere Process Server 6.0.1.x version of the `wbischedulercommon.jar` file with the WebSphere Process Server 6.1 or WebSphere Process Server 6.0.2.x version of that JARfile. You can obtain the new JAR file from the WebSphere Process Server 6.1.x *install_root*`AppScheduler/lib` directory or the WebSphere Process Server 6.0.2 *install_root*`/lib` directory. Copy the JAR file into the WebSphere Process Server 6.0.1.x `lib` directory and replace the existing JAR file. Do not rename the existing JAR file and leave it in the `lib` directory, because WebSphere Process Server picks up all files in the `lib` directory as JAR files regardless of the extension. Then, restart the server or cluster so that WebSphere Process Server picks up the new JAR file.

Handling correlation values in WebSphere Process Server

Problem: In WebSphere Process Server, new events attempting to use the existing correlation values will fail. In such instances, the error message

```
CWWBE0074E: Correlation violation in activity 'null' for correlation set
'CorrelationSetA'java.sql.
SQLException: Could not insert new row - duplicate value in a UNIQUE INDEX column
```

appears. **Cause:** When a collaboration or process instance completes in WebSphere InterChange Server or WebSphere Business Integration Server Express, data related to that instance is deleted except for cases dealing with failures. In WebSphere Process Server, the persistence of process instance-related data is controlled by the business process execution language (BPEL) option, "Automatically delete the process after completion." BPEL files generated by the WebSphere InterChange Server or WebSphere Business Integration Server Express to WebSphere Process Server migration wizard will not have this option selected. As a result, process instance data will persist, even after the process instance completes, until you clean it up manually. When a process defines a correlation set, the correlation values locked by process instances remain locked as long as the process instance data is persisted, even after the process has completed. As a result, new events attempting to use the same correlation values will fail as long as the previous process instance's data persists. This behavior will be different than in WebSphere InterChange Server or WebSphere Business Integration Server Express, where new events with duplicate correlation set values could be processed as soon as the previous instance was complete. **Solution:** To simulate the behavior of WebSphere InterChange Server or WebSphere Business Integration Server Express with respect to multiple events with duplicate correlation set values, you can choose to select the BPEL option "Automatically delete the process after completion" so that process instance data is deleted, and the correlation value is unlocked, as soon as the process instance completes. Before selecting this option, you should investigate and fully understand the way failures are handled in WebSphere Process Server and to ensure that your failed event strategy does not rely on data that will be

automatically deleted when this option is set.

Packaging and deploying migrated applications

After migrating the WebSphere InterChange Server or WebSphere Business Integration Server Express repository using the reposMigrate command, you will need to package the resulting JAR files into EAR files in order for them to be deployed to the WebSphere Process Server. To do this, you can either import each migration-generated JAR file into WebSphere Integration Developer and export the modules as EAR files, or you can use the serviceDeploy command. The serviceDeploy command accepts a JAR file as input and outputs a deployable EAR file. Packaging the migration code into EAR files involves compiling the resulting migrated JAR file. If this produces validation errors, they are most likely due to the use of unsupported WebSphere InterChange Server or WebSphere Business Integration Server Express APIs or third-party APIs that were present in WebSphere InterChange Server or WebSphere Business Integration Server Express but have not yet been included in the WebSphere Process Server class path. Remove the unsupported APIs and add the third-party classes to the WebSphere Process Server class path.

Validation errors may also be caused by not following premigration recommended practices or may indicate postmigration work that still needs to be performed on the artifacts. As with migration errors, each validation error should be handled on a per-error basis. If a recommended premigration practice was not followed, you can update the repository and migrate it again, or you can edit the output artifacts to remove the problem.

Any other validation errors should be resolved as if these artifacts were created from scratch. You should refer to the validator documentation that outlines common artifact errors and their resolutions. Inevitably an automated migration cannot account completely for your program's intent; it can just make best guesses. Therefore, even if there are no validation errors, it is possible the migrated artifacts do not perform as intended. You should review all artifacts to confirm that the intended purpose of each artifact is met by its migrated content.

Related concepts

"Limitations when migrating from WebSphere InterChange Server or WebSphere Business Integration Server Express" on page 177
Some characteristics of WebSphere InterChange Server or WebSphere Business Integration Server Express are not precisely duplicated by WebSphere Process Server. Therefore you might have to modify your applications after migration to get them to perform as they did in WebSphere InterChange Server or WebSphere Business Integration Server Express.

"Troubleshooting migration from WebSphere InterChange Server or WebSphere Business Integration Server Express" on page 179
Find solutions to problems you encounter with migration as well as instructions for turning on logging and tracing.

Related tasks

"Migrating WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts with the reposMigrate command" on page 139
Migrate WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts to WebSphere Process Server artifacts with the **reposMigrate** command.

Related reference

"Premigration considerations" on page 132
Consider these guidelines for the development of integration artifacts for WebSphere InterChange Server or WebSphere Business Integration Server Express to ease the migration of WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts to WebSphere Process Server.

Related information

 serviceDeploy command

 Wsadmin tool

 WebSphere Integration Developer information center

 IBM developerWorks

 Migrating WebSphere InterChange Server artifacts to WebSphere Process Server artifacts, Part 1: Migrating collaboration templates to BPEL

 Defining WebSphere variables

Configuring global security after WebSphere InterChange Server or WebSphere Business Integration Server Express migration

Perform these additional security configuration steps to enable projects migrated from WebSphere InterChange Server or WebSphere Business Integration Server Express to run successfully in a WebSphere Process Server environment.

Before you begin

You must first configure security for your WebSphere Process Server as described in Securing applications and their environment. In particular, make sure you have completed the steps in Securing adapters and Creating end-to-end security. In addition, install the EAR file for each module. Refer to Deploying (installing) secure applications for details.

About this task

After performing the above tasks, you are ready to complete the configuration steps, as follows:

- Binding the message-driven bean to activation specification
- Mapping the resource references to resources
- Mapping security roles to users or groups (required only when monitoring Common Based Events)
- Mapping RunAs roles (required only when monitoring Common Based Events)

Note: Mapping security roles to users or groups and mapping RunAs roles is possible from the administrative console only if the EJB deployment descriptors for the EJB projects have had a RunAs role defined. See Mapping users to RunAs roles using an assembly tool in the WebSphere Application Server Network Deployment, version 6.1 information center for information about defining RunAs roles with an assembly tool.

Procedure

1. Bind the message-driven bean to activation specification.
   a. From the administrative console, select **Applications > Enterprise Applications**.
   b. In the right panel, select the name of the application you just installed. (Select the name, not the check box to the left of the name.)
   c. In the right panel again, under Enterprise Java Bean Properties, select **Message Driven Bean listener bindings**.
   d. For each import or export EJB (indicated by an EJB name that starts with "_import" or "_export"), under the Bindings column, specify **SCA_Auth_Alias** in the "ActivationSpec authentication alias" field.
   e. Select **OK**, then **Save**.
2. Map the resource references to resources.
   a. From the administrative console, select **Applications > Enterprise Applications**.
   b. In the right panel, select the name of the application you just installed. (Select the name, not the check box to the left of the name.)
   c. In the right panel, under References, select **Resource references**.
   d. In the Specify authentication method: field under javax.jms.ConnectionFactory, select the **Use default method (many to one mapping)** radio button.
   e. In the **Select authentication data entry** pull down menu, select **SCA_Auth_Alias**.
   f. Check the check box to select all of the modules.
   g. Select **Apply**, then **OK**, then **Save**.
3. Map security roles to user groups.
   a. From the administrative console, select **Applications > Enterprise Applications**.
   b. In the right panel, select the name of the application you just installed. (Select the name, not the check box to the left of the name.)
   c. In the right panel, under Detail Properties, select **Security role to user/group mapping**.

d. Select the check box to the left of the role you want to map and then select **Look up users**.
e. Select **Search** to display a list of users who are available to map to the role, and move the correct user name to the "Selected:" column.
f. Select **OK**. The "Security role to user/group mapping" panel will be redisplayed.
g. Uncheck the check boxes in the "Everyone?" and "All authenticated?" columns corresponding to the role, and select **OK**, then **Save**.

4. Map RunAs roles.
   a. From the administrative console, select **Applications > Enterprise Applications**.
   b. In the right panel, select the name of the application you just installed. (Select the name, not the check box to the left of the name.)
   c. In the right panel, under Detail Properties, select **User RunAs roles**.
   d. Select the check box next to the role you mapped in step 3 on page 154.
   e. Enter the user name and password corresponding to the user name selected in step 3e into the **Username** and **Password** fields, respectively.
   f. Select **Apply**.
   g. Select **OK**, then **Save**.

What to do next

After installing and configuring all of the EAR projects, select **Applications > Enterprise Applications** in the administrative console and start the installed migrated projects. If they start successfully, then you are now ready to send events through one of the inbound connectors to be processed by the server.

Related information

Wsadmin tool

WebSphere InterChange Server v4.3 documentation

Mapping users to RunAs roles using an assembly tool

Securing adapters

Securing applications and their environment

Deploying (installing) secure applications

Creating end to end security

Support for WebSphere Business Integration data handlers

The data handler support API enables certain data handler methods to be invoked from the AccessEJB, a WebSphere Process Server SCA Java component, or WebSphere Process Server bindings.

WebSphere Process Server (version 6.0.2.3 and higher) provides a data handler support Application Programming Interface (API) which enables select WebSphere Business Integration data handler methods to be invoked from the AccessEJB, a WebSphere Process Server SCA Java component, or WebSphere Process Server bindings. Access EJB has been replicated as an EJB that allows JService calls to route the input business object to the appropriate migrated module. The BPEL file

in the migrated module will be invoked instead of the original WebSphere InterChange Server or WebSphere Business Integration Server Express target collaboration.

WebSphere Process Server bindings invoke data bindings to perform data transformation.WebSphere Process Server provides several built-in data bindings as well as the capability to provide user-defined data bindings. You can implement a user-defined, or custom data binding to invoke a WebSphere Business Integration data handler.

By providing a custom data binding implementation, it is possible to leverage WebSphere Business Integration data handlers via the data handler support API. The data handler support API provides wrapper methods around existing WebSphere Business Integration data handler interface methods which perform the conversion between WebSphere Business Integration business objects and SDOs.

Data handler support API

By providing a custom data binding implementation, it is possible to leverage WebSphere Business Integration data handlers via the data handler support API. This API defines a set of public methods which can be invoked from a custom data binding or a Java component. It provides a way to invoke a text-based WebSphere Business Integration data handler from a process server binding. The following are the API methods:

`getSDOFromString(String inputString, String sdoName, String metaObjectName, String mimeType)` (Returns `dataObject`)

`getStringFromSDO(DataObject sdo, String metaObjectName, String mimeType)` (Returns `String`)

You can access these methods with the Java class `com.ibm.wbi.datahandler.JavaConnectorUtilDH`. This is the class that IBM recommends that you use from a data binding or a Java component. If you have existing code you can use the `AppSide_Connector.JavaConnectorUtil` class.

Usage

The methods defined in the data handler support API can be invoked from either a WebSphere Process Server binding or a Java component. However, because data is typically transformed in the binding in a WebSphere Process Server environment, IBM strongly recommends that the methods of the data handler support API be invoked from a custom data binding rather than a Java component.

Limitations

The data handler support API has the following limitations:

- Binary conversion methods are not supported. That is, no support is provided for getByteArrayFromSDO(), getStreamFromSDO(), getSDO(byte[], and similar calls.
- setEncoding(), setLocale() and setOptions() methods are not exposed via the data handler support API.
- Dynamic child meta objects are not supported.
- You must use WebSphere Business Integration Adapter business object tooling for creation of new objects.

Related reference

"Supported WebSphere InterChange Server or WebSphere Business Integration Server Express APIs"
In addition to the WebSphere InterChange Server or WebSphere Business Integration Server Express source artifact migration tools provided in WebSphere Process Server and WebSphere Integration Developer, WebSphere Process Server also provides support for many of the APIs that were provided in WebSphere InterChange Server or WebSphere Business Integration Server Express. The migration tools work in conjunction with these WebSphere InterChange Server or WebSphere Business Integration Server Express APIs by preserving your custom snippet code as much as possible when migrating.

Related information

 IBM WebSphere Business Integration Adapters/IBM WebSphere InterChange Server Data Handler Guide

Supported WebSphere InterChange Server or WebSphere Business Integration Server Express APIs

In addition to the WebSphere InterChange Server or WebSphere Business Integration Server Express source artifact migration tools provided in WebSphere Process Server and WebSphere Integration Developer, WebSphere Process Server also provides support for many of the APIs that were provided in WebSphere InterChange Server or WebSphere Business Integration Server Express. The migration tools work in conjunction with these WebSphere InterChange Server or WebSphere Business Integration Server Express APIs by preserving your custom snippet code as much as possible when migrating.

Note: These APIs are provided only to support migrated WebSphere InterChange Server or WebSphere Business Integration Server Express applications until they can be modified to use new WebSphere Process Server APIs.
The supported WebSphere InterChange Server or WebSphere Business Integration Server Express APIs are listed below. These APIs provide functions in WebSphere Process Server similar to the function that they provide in WebSphere InterChange Server or WebSphere Business Integration Server Express. See the WebSphere InterChange Server or WebSphere Business Integration Server Express v4.3 documentation for a functional description of these APIs.

CwBiDiEngine
AppSide_Connector/

- BiDiBOTransformation(BusinessObject, String, String, boolean):BusinessObj
- BiDiBusObjTransformation(BusObj, String, String, boolean):BusObj
- BiDiStringTransformation(String, String, String):String

JavaConnectorUtil
AppSide_Connector/

- INFRASTRUCTURE_MESSAGE_FILE
- CONNECTOR_MESSAGE_FILE
- XRD_WARNING
- XRD_TRACE
- XRD_INFO
- XRD_ERROR
- XRD_FATAL

- LEVEL1
- LEVEL2
- LEVEL3
- LEVEL4
- LEVEL5
- createBusinessObject(String):BusinesObjectInterface
- createBusinessObject(String, Locale):BusinesObjectInterface
- createBusinessObject(String, String):BusinesObjectInterface
- createContainer(String):CxObjectContainerInterface
- generateMsg(int, int, int, int, int, Vector):String
- generateMsg(int, int, int, int, Vector):String
- getBlankValue():String
- getEncoding():String
- getIgnoreValue():String
- getLocale():String
- getSDOFromString(String inputString, String sdoName, String metaObjectName, String mimeType)
- getStringFromSDO(DataObject sdo, String metaObjectName, String mimeType)
- isBlankValue(Object):boolean
- isIgnoreValue(Object):boolean
- isTraceEnabled(int):boolean
- logMsg(String)
- logMsg(String, int)
- traceWrite(int, String)

JavaConnectorUtilDH
datahandler/
wbi/
ibm/
com/

- getSDOFromString(String inputString, String sdoName, String metaObjectName, String mimeType)
- getStringFromSDO(DataObject sdo, String metaObjectName, String mimeType)

BusObj
Collaboration/

- BusObj(DataObject)
- BusObj(String)
- BusObj(String, Locale)
- copy(BusObj)
- duplicate():BusObj
- equalKeys(BusObj):boolean
- equals(Object):boolean
- equalsShallow(BusObj):boolean
- exists(String):boolean
- get(int):Object
- get(String):Object

- getBoolean(String):boolean
- getBusObj(String):BusObj
- getBusObjArray(String):BusObjArray
- getCount(String):int
- getDouble(String):double
- getFloat(String):float
- getInt(String):int
- getKeys():String
- getLocale():java.util.Locale
- getLong(String):long
- getLongText(String):String
- getString(String):String
- getType():String
- getValues():String
- getVerb():String
- isBlank(String):boolean
- isKey(String):boolean
- isNull(String):boolean
- isRequired(String):boolean
- keysToString():String
- set(BusObj)
- set(int, Object)
- set(String, boolean)
- set(String, double)
- set(String, float)
- set(String, int)
- set(String, long)
- set(String, Object)
- set(String, String)
- setContent(BusObj)
- setDefaultAttrValues()
- setKeys(BusObj)
- setLocale(java.util.Locale)
- setVerb(String)
- setVerbWithCreate(String, String)
- setWithCreate(String, boolean)
- setWithCreate(String, BusObj)
- setWithCreate(String, BusObjArray)
- setWithCreate(String, double)
- setWithCreate(String, float)
- setWithCreate(String, int)
- setWithCreate(String, long):
- setWithCreate(String, Object)
- setWithCreate(String, String)
- toString():String

- validData(String, boolean):boolean
- validData(String, BusObj):boolean
- validData(String, BusObjArray):boolean
- validData(String, double):boolean
- validData(String, float):boolean
- validData(String, int):boolean
- validData(String, long):boolean
- validData(String, Object):boolean
- validData(String, String):boolean

BusObjArray
Collaboration/

- addElement(BusObj)
- duplicate():BusObjArray
- elementAt(int):BusObj
- equals(BusObjArray):boolean
- getElements():BusObj[]
- getLastIndex():int
- max(String):String
- maxBusObjArray(String):BusObjArray
- maxBusObjs(String):BusObj[]
- min(String):String
- minBusObjArray(String):BusObjArray
- minBusObjs(String):BusObj[]
- removeAllElements()
- removeElement(BusObj)
- removeElementAt(int)
- setElementAt(int, BusObj)
- size():int
- sum(String):double
- swap(int, int)
- toString():String

BaseDLM
DLM/

- BaseDLM(BaseMap)
- getDBConnection(String):CwDBConnection
- getDBConnection(String, boolean):CwDBConnection
- getName():String
- getRelConnection(String):DtpConnection
- implicitDBTransactionBracketing():boolean
- isTraceEnabled(int):boolean
- logError(int)
- logError(int, Object[])
- logError(int, String)
- logError(int, String, String)

- logError(int, String, String, String)
- logError(int, String, String, String, String)
- logError(int, String, String, String, String, String)
- logError(String)
- logInfo(int)
- logInfo(int, Object[])
- logInfo(int, String)
- logInfo(int, String, String)
- logInfo(int, String, String, String)
- logInfo(int, String, String, String, String)
- logInfo(int, String, String, String, String, String)
- logInfo(String)
- logWarning(int)
- logWarning(int, Object[])
- logWarning(int, String)
- logWarning(int, String, String)
- logWarning(int, String, String, String)
- logWarning(int, String, String, String, String)
- logWarning(int, String, String, String, String, String)
- logWarning(String)
- raiseException(RunTimeEntityException)
- raiseException(String, int)
- raiseException(String, int, Object[])
- raiseException(String, int, String)
- raiseException(String, int, String, String)
- raiseException(String, int, String, String, String)
- raiseException(String, int, String, String, String, String)
- raiseException(String, int, String, String, String, String, String)
- raiseException(String, String)
- releaseRelConnection(boolean)
- trace(int, int)
- trace(int, int, Object[])
- trace(int, int, String)
- trace(int, int, String, String)
- trace(int, int, String, String, String)
- trace(int, int, String, String, String, String)
- trace(int, int, String, String, String, String, String)
- trace(int, String)
- trace(String)

CwDBConnection
CwDBConnection/
CxCommon/

- beginTransaction()
- commit()
- executePreparedSQL(String)

- executePreparedSQL(String, Vector)
- executeSQL(String)
- executeSQL(String, Vector)
- executeStoredProcedure(String, Vector)
- getUpdateCount():int
- hasMoreRows():boolean
- inTransaction():boolean
- isActive():boolean
- nextRow():Vector
- release()
- rollback()

CwDBConstants
CwDBConnection/
CxCommon/

- PARAM_IN - 0
- PARAM_INOUT - 1
- PARAM_OUT - 2

CwDBStoredProcedureParam
CwDBConnection/
CxCommon/

- CwDBStoredProcedureParam(int, Array)
- CwDBStoredProcedureParam(int, BigDecimal)
- CwDBStoredProcedureParam(int, boolean)
- CwDBStoredProcedureParam(int, Boolean)
- CwDBStoredProcedureParam(int, byte[])
- CwDBStoredProcedureParam(int, double)
- CwDBStoredProcedureParam(int, Double)
- CwDBStoredProcedureParam(int, float)
- CwDBStoredProcedureParam(int, Float)
- CwDBStoredProcedureParam(int, int)
- CwDBStoredProcedureParam(int, Integer)
- CwDBStoredProcedureParam(int, java.sql.Blob)
- CwDBStoredProcedureParam(int, java.sql.Clob)
- CwDBStoredProcedureParam(int, java.sql.Date)
- CwDBStoredProcedureParam(int, java.sql.Struct)
- CwDBStoredProcedureParam(int, java.sql.Time)
- CwDBStoredProcedureParam(int, java.sql.Timestamp)
- CwDBStoredProcedureParam(int, Long)
- CwDBStoredProcedureParam(int, String)
- CwDBStoredProcedureParam(int, String, Object)
- getParamType():int getValue():Object

DataHandler (Abstract Class)
DataHandlers/
crossworlds/
com/

- createHandler(String, String, String):DataHandler
- getBO(InputStream, Object):BusinessObjectInterface
- getBO(Object, BusinessObjectInterface, Object)
- getBO(Object, Object):BusinessObjectInterface
- getBO(Reader, BusinessObjectInterface, Object) (Abstract Method)
- getBO(Reader, Object):BusinessObjectInterface (Abstract Method)
- getBO(String, Object):BusinessObjectInterface
- getBOName(InputStream):String
- getBOName(Reader):String
- getBOName(String):String
- getBooleanOption(String):boolean
- getEncoding():String
- getLocale():Locale
- getOption(String):String
- getStreamFromBO(BusinessObjectInterface, Object):InputStream (Abstract Method)
- getStringFromBO(BusinessObjectInterface, Object):String (Abstract Method)
- setConfigMOName(String)
- setEncoding(String)
- setLocale(Locale)
- setOption(String, String)
- traceWrite(String, int)

NameHandler (Abstract Class)
DataHandlers/
crossworlds/
com/

- getBOName(Reader, String):String) (Abstract Method)

ConfigurationException (extends java.lang.Exception)
Exceptions/
DataHandlers/
crossworlds/
com/

MalformedDataException (extends java.lang.Exception)
Exceptions/
DataHandlers/
crossworlds/
com/

NotImplementedException (extends java.lang.Exception)
Exceptions/
DataHandlers/
crossworlds/
com/

BusinessObjectInterface
CxCommon/

- clone():Object
- dump():String
- getAppText():String
- getAttrCount():int
- getAttrDesc(int):CxObjectAttr
- getAttrDesc(String):CxObjectAttr
- getAttribute(String):Object
- getAttributeIndex(String):int
- getAttributeType(int):int
- getAttributeType(String):int
- getAttrName(int):String
- getAttrValue(int):Object
- getAttrValue(String):Object
- getBusinessObjectVersion():String
- getDefaultAttrValue(int):String
- getDefaultAttrValue(String):String
- getLocale():String
- getName():String
- getParentBusinessObject():BusinessObjectInterface
- getVerb():String
- getVerbAppText(String):String
- isBlank(int):boolean
- isBlank(String):boolean
- isIgnore(int):boolean
- isIgnore(String):boolean
- isVerbSupported(String):boolean
- makeNewAttrObject(int):Object
- makeNewAttrObject(String):Object
- setAttributeWithCreate(String, Object)
- setAttrValue(int, Object)
- setAttrValue(String, Object)
- setDefaultAttrValues()
- setLocale(Locale)
- setLocale(String)
- setVerb(String)

CxObjectAttr
CxCommon/

- BOOLEAN
- BOOLSTRING
- DATE
- DATESTRING
- DOUBLE
- DOUBSTRING

- FLOAT
- FLTSTRING
- INTEGER
- INTSTRING
- INVALID_TYPE_NUM
- INVALID_TYPE_STRING
- LONGTEXT
- LONGTEXTSTRING
- MULTIPLECARDSTRING
- OBJECT
- SINGLECARDSTRING
- STRING
- STRSTRING
- equals(Object):boolean
- getAppText():String
- getCardinality():String
- getDefault():String
- getMaxLength():int
- getName():String
- getRelationType():String
- getTypeName():String
- getTypeNum():String
- hasCardinality(String):boolean
- hasName(String):boolean
- hasType(String):boolean
- isForeignKeyAttr():boolean
- isKeyAttr():boolean
- isMultipleCard():boolean
- isObjectType():boolean
- isRequiredAttr():boolean
- isType(Object):boolean

CxObjectContainerInterface
CxCommon/

- getBusinessObject(int):BusinessObjectInterface
- getObjectCount():int
- insertBusinessObject(BusinessObjectInterface)
- removeAllObjects()
- removeBusinessObjectAt(int)
- setBusinessObject(int, BusinessObjectInterface)

DtpConnection
Dtp/
CxCommon/

- beginTran()
- commit()

- executeSQL(String)
- executeSQL(String, Vector)
- executeStoredProcedure(String, Vector)
- getUpdateCount():int
- hasMoreRows():boolean
- inTransaction():boolean
- isActive():boolean
- nextRow():Vector
- rollback()

DtpDataConversion
Dtp/
CxCommon/

- BOOL_TYPE - 4
- CANNOTCONVERT - 2
- DATE_TYPE - 5
- DOUBLE_TYPE - 3
- FLOAT_TYPE - 2
- INTEGER_TYPE - 0
- LONGTEXT_TYPE - 6
- OKTOCONVERT - 0
- POTENTIALDATALOSS - 1
- STRING_TYPE - 1
- UNKNOWN_TYPE - 999
- getType(double):int
- getType(float):int
- getType(int):int
- getType(Object):int
- isOKToConvert(int, int):int
- isOKToConvert(String, String):int
- toBoolean(boolean):Boolean
- toBoolean(Object):Boolean
- toDouble(double):Double
- toDouble(float):Double
- toDouble(int):Double
- toDouble(Object):Double
- toFloat(double):Float
- toFloat(float):Float
- toFloat(int):Float
- toFloat(Object):Float
- toInteger(double):Integer
- toInteger(float):Integer
- toInteger(int):Integer
- toInteger(Object):Integer
- toPrimitiveBoolean(Object):boolean
- toPrimitiveDouble(float):double

- toPrimitiveDouble(int):double
- toPrimitiveDouble(Object):double
- toPrimitiveFloat(double):float
- toPrimitiveFloat(int):float
- toPrimitiveFloat(Object):float
- toPrimitiveInt(double):int
- toPrimitiveInt(float):int
- toPrimitiveInt(Object):int
- toString(double):String
- toString(float):String
- toString(int):String
- toString(Object):String

DtpDate
Dtp/
CxCommon/

- DtpDate()
- DtpDate(long, boolean)
- DtpDate(String, String)
- DtpDate(String, String, String[], String[])
- addDays(int):DtpDate
- addMonths(int):DtpDate
- addWeekdays(int):DtpDate
- addYears(int):DtpDate
- after(DtpDate):boolean
- before(DtpDate):boolean
- calcDays(DtpDate):int
- calcWeekdays(DtpDate):int
- get12MonthNames():String[]
- get12ShortMonthNames():String[]
- get7DayNames():String[]
- getCWDate():String
- getDayOfMonth():String
- getDayOfWeek():String
- getHours():String
- getIntDay():int
- getIntDayOfWeek():int
- getIntHours():int
- getIntMilliSeconds():int
- getIntMinutes():int
- getIntMonth():int
- getIntSeconds():int
- getIntYear():int
- getMaxDate(BusObjArray, String, String):DtpDate
- getMaxDateBO(BusObj[], String, String):BusObj[]
- getMaxDateBO(BusObjArray, String, String):BusObj[]

- getMinDate(BusObjArray, String, String):DtpDate
- getMinDateBO(BusObj[], String, String):BusObj[]
- getMinDateBO(BusObjArray, String, String):BusObj[]
- getMinutes():String
- getMonth():String
- getMSSince1970():long
- getNumericMonth():String
- getSeconds():String
- getShortMonth():String
- getYear():String
- set12MonthNames(String[], boolean)
- set12MonthNamesToDefault()
- set12ShortMonthNames(String[])
- set12ShortMonthNamesToDefault()
- set7DayNames(String[])
- set7DayNamesToDefault()
- toString():String
- toString(String):String
- toString(String, boolean):String

DtpMapService
Dtp/
CxCommon/

- runMap(String, String, BusObj[], CxExecutionContext):BusObj[]

DtpSplitString
Dtp/
CxCommon/

- DtpSplitString(String, String)
- elementAt(int):String
- firstElement():String
- getElementCount():int
- getEnumeration():Enumeration
- lastElement():String
- nextElement():String
- prevElement():String
- reset()

DtpUtils
Dtp/
CxCommon/

- padLeft(String, char, int):String
- padRight(String, char, int):String
- stringReplace(String, String, String):String
- truncate(double):int
- truncate(double, int):double
- truncate(float):int

- truncate(float, int):double
- truncate(Object):int
- truncate(Object, int):double

BusObjInvalidVerbException (extends InterchangeExceptions)
Exceptions/
CxCommon/

- getFormattedMessage()

IdentityRelationship
relationship/
utilities/
crossworlds/
com/

- addMyChildren(String, String, BusObj, String, Object, CxExecutionContext)
- deleteMyChildren(String, String, BusObj, String, CxExecutionContext)
- deleteMyChildren(String, String, BusObj, String, Object, CxExecutionContext)
- foreignKeyLookup(String, String, BusObj, String, BusObj, String, CxExecutionContext)
- foreignKeyXref(String, String, String, BusObj, String, BusObj, String, CxExecutionContext)
- maintainChildVerb(String, String, String, BusObj, String, BusObj, String, CxExecutionContext, boolean, boolean)
- maintainCompositeRelationship(String, String, BusObj, Object, CxExecutionContext)
- maintainSimpleIdentityRelationship(String, String, BusObj, BusObj, CxExecutionContext)
- updateMyChildren(String, String, BusObj, String, String, String, String, CxExecutionContext)

MapExeContext
Dtp/
CxCommon/

- ACCESS_REQUEST - "SUBSCRIPTION_DELIVERY"
- ACCESS_RESPONSE - "ACCESS_RETURN_REQUEST"
- EVENT_DELIVERY - "SUBSCRIPTION_DELIVERY"
- SERVICE_CALL_FAILURE - "CONSUME_FAILED"
- SERVICE_CALL_REQUEST - "CONSUME"
- SERVICE_CALL_RESPONSE - "DELIVERBUSOBJ"
- getConnName():String
- getGenericBO():BusObj
- getInitiator():String
- getLocale():java.util.Locale
- getOriginalRequestBO():BusObj
- setConnName(String)
- setInitiator(String)
- setLocale(java.util.Locale)

Participant
RelationshipServices/
Server/

- Participant(String, String, int, BusObj)
- Participant(String, String, int, String)
- Participant(String, String, int, long)
- Participant(String, String, int, int)
- Participant(String, String, int, double)
- Participant(String, String, int, float)
- Participant(String, String, int, boolean)
- Participant(String, String, BusObj)
- Participant(String, String, String)
- Participant(String, String, long)
- Participant(String, String, int)
- Participant(String, String, double)
- Participant(String, String, float)
- Participant(String, String, boolean)
- getBoolean():boolean
- getBusObj():BusObj
- getDouble():double
- getFloat():float
- getInstanceId():int
- getInt():int
- getLong():long
- getParticipantDefinition():String
- getRelationshipDefinition():String
- getString():String INVALID_INSTANCE_ID
- set(boolean)
- set(BusObj)
- set(double)
- set(float)
- set(int)
- set(long)
- set(String)
- setInstanceId(int)
- setParticipantDefinition(String)
- setRelationshipDefinition(String)
- setParticipantDefinition(String)
- setRelationshipDefinition(String)

Relationship
RelationshipServices/
Server/

- addMyChildren(String, String, BusObj, String, Object, CxExecutionContext)
- addParticipant(Participant):int
- addParticipant(String, String, boolean):int

- addParticipant(String, String, BusObj):int
- addParticipant(String, String, double):int
- addParticipant(String, String, float):int
- addParticipant(String, String, int):int
- addParticipant(String, String, int, boolean):int
- addParticipant(String, String, int, BusObj):int
- addParticipant(String, String, int, double):int
- addParticipant(String, String, int, float):int
- addParticipant(String, String, int, int):int
- addParticipant(String, String, int, long):int
- addParticipant(String, String, int, String):int
- addParticipant(String, String, long):int
- addParticipant(String, String, String):int
- create(Participant):int
- create(String, String, boolean):int
- create(String, String, BusObj):int
- create(String, String, double):int
- create(String, String, float):int
- create(String, String, int):int
- create(String, String, long):int
- create(String, String, String):int
- deactivateParticipant(Participant)
- deactivateParticipant(String, String, boolean)
- deactivateParticipant(String, String, BusObj)
- deactivateParticipant(String, String, double)
- deactivateParticipant(String, String, float)
- deactivateParticipant(String, String, int)
- deactivateParticipant(String, String, long)
- deactivateParticipant(String, String, String)
- deactivateParticipantByInstance(String, String, int)
- deactivateParticipantByInstance(String, String, int, boolean)
- deactivateParticipantByInstance(String, String, int, BusObj)
- deactivateParticipantByInstance(String, String, int, double)
- deactivateParticipantByInstance(String, String, int, float)
- deactivateParticipantByInstance(String, String, int, int)
- deactivateParticipantByInstance(String, String, int, long)
- deactivateParticipantByInstance(String, String, int, String)
- deleteMyChildren(String, String, BusObj, String, CxExecutionContext)
- deleteMyChildren(String, String, BusObj, String, Object, CxExecutionContext)
- deleteParticipant(Participant)
- deleteParticipant(String, String, boolean)
- deleteParticipant(String, String, BusObj)
- deleteParticipant(String, String, double)
- deleteParticipant(String, String, float)
- deleteParticipant(String, String, int)

- deleteParticipant(String, String, long)
- deleteParticipant(String, String, String)
- deleteParticipantByInstance(String, String, int)
- deleteParticipantByInstance(String, String, int, boolean)
- deleteParticipantByInstance(String, String, int, BusObj)
- deleteParticipantByInstance(String, String, int, double)
- deleteParticipantByInstance(String, String, int, float)
- deleteParticipantByInstance(String, String, int, int)
- deleteParticipantByInstance(String, String, int, long)
- deleteParticipantByInstance(String, String, int, String)
- getNewID(String):int
- maintainCompositeRelationship(String, String, BusObj, Object, CxExecutionContext)
- maintainSimpleIdentityRelationship(String, String, BusObj, BusObj, CxExecutionContext)
- retrieveInstances(String, boolean):int[]
- retrieveInstances(String, BusObj):int[]
- retrieveInstances(String, double):int[]
- retrieveInstances(String, float):int[]
- retrieveInstances(String, int):int[]
- retrieveInstances(String, long):int[]
- retrieveInstances(String, String):int[]
- retrieveInstances(String, String, boolean):int[]
- retrieveInstances(String, String, BusObj):int[]
- retrieveInstances(String, String, double):int[]
- retrieveInstances(String, String, float):int[]
- retrieveInstances(String, String, int):int[]
- retrieveInstances(String, String, long):int[]
- retrieveInstances(String, String, String):int[]
- retrieveInstances(String, String[], boolean):int[]
- retrieveInstances(String, String[], BusObj):int[]
- retrieveInstances(String, String[], double):int[]
- retrieveInstances(String, String[], float):int[]
- retrieveInstances(String, String[], int):int[]
- retrieveInstances(String, String[], long):int[]
- retrieveInstances(String, String[], String):int[]
- retrieveParticipants(String):Participant[]
- retrieveParticipants(String, String):Participant[]
- retrieveParticipants(String, String[]):Participant[]
- retrieveParticipants(String, int):Participant[]
- retrieveParticipants(String, String, int):Participant[]
- retrieveParticipants(String, String[], int):Participant[]
- updateMyChildren(String, String, BusObj, String, String, String, String, CxExecutionContext)
- updateParticipant(String, String, BusObj)

- updateParticipantByInstance(Participant)
- updateParticipantByInstance(String, String, int)
- updateParticipantByInstance(String, String, int, BusObj)

UserStoredProcedureParam
Dtp/
CxCommon/
- UserStoredProcedureParam(int, String, Object, String, String)
- getParamDataTypeJavaObj():String
- getParamDataTypeJDBC():int
- getParamIndex():int
- getParamIOType():String
- getParamName():String
- getParamValue():Object
- setParamDataTypeJavaObj(String)
- setParamDataTypeJDBC(int)
- setParamIndex(int)
- setParamIOType(String)
- setParamName(String)
- setParamValue(Object)
- PARAM_TYPE_IN - "IN"
- PARAM_TYPE_OUT - "OUT"
- PARAM_TYPE_INOUT - "INOUT"
- DATA_TYPE_STRING - "String"
- DATA_TYPE_INTEGER - "Integer"
- DATA_TYPE_DOUBLE - "Double"
- DATA_TYPE_FLOAT - "Float"
- DATA_TYPE_BOOLEAN - "Boolean"
- DATA_TYPE_TIME - "java.sql.Time"
- DATA_TYPE_DATE - "java.sql.Date"
- DATA_TYPE_TIMESTAMP - "java.sql.Timestamp"
- DATA_TYPE_BIG_DECIMAL - "java.math.BigDecimal"
- DATA_TYPE_LONG_INTEGER - "Long"
- DATA_TYPE_BINARY - "byte[]"
- DATA_TYPE_CLOB - "Clob"
- DATA_TYPE_BLOB - "Blob"
- DATA_TYPE_ARRAY - "Array"
- DATA_TYPE_STRUCT - "Struct"
- DATA_TYPE_REF - "Ref"

BaseCollaboration
Collaboration/
- BaseCollaboration(com.ibm.bpe.api.ProcessInstanceData)
- AnyException - "AnyException"
- AppBusObjDoesNotExist - "BusObjDoesNotExist"
- AppLogOnFailure - "AppLogOnFailure"

- AppMultipleHits - "AppMultipleHits"
- AppRequestNotYetSent - "AppRequestNotYetSent"
- AppRetrieveByContentFailed - "AppRetrieveByContent"
- AppTimeOut - "AppTimeOut"
- AppUnknown - "AppUnknown"
- AttributeException - "AttributeException"
- existsConfigProperty(String):boolean
- getConfigProperty(String):String
- getConfigPropertyArray(String):String[]
- getCurrentLoopIndex():int
- getDBConnection(String):CwDBConnection
- getDBConnection(String, boolean):CwDBConnection getLocale():java.util.Locale
- getMessage(int):String
- getMessage(int, Object[]):String
- getName():String
- implicitDBTransactionBracketing():boolean
- isCallerInRole(String):boolean
- isTraceEnabled(int):boolean
- JavaException - "JavaException"
- logError(int)
- logError(int, Object[])
- logError(int, String)
- logError(int, String, String)
- logError(int, String, String, String)
- logError(int, String, String, String, String)
- logError(int, String, String, String, String, String)
- logError(String)
- logInfo(int)
- logInfo(int, Object[])
- logInfo(int, String)
- logInfo(int, String, String)
- logInfo(int, String, String, String)
- logInfo(int, String, String, String, String)
- logInfo(int, String, String, String, String, String)
- logInfo(String)
- logWarning(int)
- logWarning(int, Object[])
- logWarning(int, String)
- logWarning(int, String, String)
- logWarning(int, String, String, String)
- logWarning(int, String, String, String, String)
- logWarning(int, String, String, String, String, String)
- logWarning(String)
- not(boolean):boolean ObjectException - "ObjectException"
- OperationException - "OperationException"

- raiseException(CollaborationException)
- raiseException(String, int)
- raiseException(String, int, Object[])
- raiseException(String, int, String)
- raiseException(String, int, String, String)
- raiseException(String, int, String, String, String)
- raiseException(String, int, String, String, String, String)
- raiseException(String, int, String, String, String, String, String)
- raiseException(String, String)
- ServiceCallException - "ConsumerException"
- ServiceCallTransportException - "ServiceCallTransportException"
- SystemException - "SystemException"
- trace(int, int)
- trace(int, int, Object[])
- trace(int, int, String)
- trace(int, int, String, String)
- trace(int, int, String, String, String)
- trace(int, int, String, String, String, String)
- trace(int, int, String, String, String, String, String)
- trace(int, String)
- trace(String)
- TransactionException - "TransactionException"

CxExecutionContext
CxCommon/
- CxExecutionContext()
- getContext(String):Object
- MAPCONTEXT - "MAPCONTEXT"
- setContext(String, Object)

CollaborationException
Collaboration/
- getMessage():String
- getMsgNumber():int
- getSubType():String
- getText():String
- getType():String
- toString():String

Filter
crossworlds/
com/
- Filter(BaseCollaboration)
- filterExcludes(String, String):boolean
- filterIncludes(String, String):boolean
- recurseFilter(BusObj, String, boolean, String, String):boolean
- recursePreReqs(String, Vector):int

Globals
crossworlds/
com/

- Globals(BaseCollaboration)
- callMap(String, BusObj):BusObj

SmartCollabService
crossworlds/
com/

- SmartCollabService()
- SmartCollabService(BaseCollaboration)
- doAgg(BusObj, String, String, String):BusObj
- doMergeHash(Vector, String, String):Vector
- doRecursiveAgg(BusObj, String, String, String):BusObj
- doRecursiveSplit(BusObj, String):Vector
- doRecursiveSplit(BusObj, String, boolean):Vector
- getKeyValues(BusObj, String):String
- merge(Vector, String):BusObj
- merge(Vector, String, BusObj):BusObj
- split(BusObj, String):Vector

StateManagement
crossworlds/
com/

- StateManagement()
- beginTransaction()
- commit()
- deleteBO(String, String, String)
- deleteState(String, String, String, int)
- persistBO(String, String, String, String, BusObj)
- recoverBO(String, String, String):BusObj
- releaseDBConnection()
- resetData()
- retrieveState(String, String, String, int):int
- saveState(String, String, String, String, int, int, double)
- setDBConnection(CwDBConnection)
- updateBO(String, String, String, String, BusObj)
- updateState(String, String, String, String, int, int)

EventKeyAttrDef
EventManagement/
CxCommon/

- EventKeyAttrDef()
- EventKeyAttrDef(String, String)
- public String keyName
- public String keyValue

EventQueryDef
EventManagement/
CxCommon/

- EventQueryDef()
- EventQueryDef(String, String, String, String, int)
- public String nameConnector
- public String nameCollaboration
- public String nameBusObj
- public String verb
- public int ownerType

FailedEventInfo
EventManagement/
CxCommon/

- FailedEventInfo()
- FailedEventInfo(String x6, int, EventKeyAttrDef[], int, int, String, String, int)
- public String nameOwner
- public String nameConnector
- public String nameBusObj
- public String nameVerb
- public String strTime
- public String strMessage
- public int wipIndex
- public EventKeyAttrDef[] strbusObjKeys
- public int nKeys
- public int eventStatus
- public String expirationTime
- public String scenarioName
- public int scenarioState

Limitations when migrating from WebSphere InterChange Server or WebSphere Business Integration Server Express

Some characteristics of WebSphere InterChange Server or WebSphere Business Integration Server Express are not precisely duplicated by WebSphere Process Server. Therefore you might have to modify your applications after migration to get them to perform as they did in WebSphere InterChange Server or WebSphere Business Integration Server Express.

The following sections describe these limitations and possible solutions.

Transaction levels

There is no direct mapping of the levels of transaction between WebSphere InterChange Server or WebSphere Business Integration Server Express collaborations and WebSphere Process Server BPEL files. Therefore, the transaction level specified in the WebSphere InterChange Server or WebSphere Business Integration Server Express collaboration is ignored and the default BPEL transaction level will be used in the migrated application. You should understand BPEL transactions and adjust your migrated applications accordingly to get the desired functionality.

Note: Pending transactions will not be migrated. All transactions should be concluded before starting migration.

Compensation

WebSphere Process Server compensation is different from WebSphere InterChange Server or WebSphere Business Integration Server Express compensation. You should evaluate the new types of compensation offered by WebSphere Process Server and choose the type that best suits your application.

Event Summary and Change Summary not supported when using WebSphere InterChange Server or WebSphere Business Integration Server Express APIs on WebSphere Process Server

Problem: Event Summary and Change Summary do not contain expected information in migrated WebSphere InterChange Server or WebSphere Business Integration Server Express applications. **Cause:** Business Objects (BusObjs) in WebSphere InterChange Server or WebSphere Business Integration Server Express do not support Change Summary and Event Summary. The WebSphere InterChange Server or WebSphere Business Integration Server Express APIs supported in WebSphere Process Server work with the WebSphere InterChange Server or WebSphere Business Integration Server Express type BusObj, so any use of those APIs forces a conversion into a BusObj. When this happens, any Event Summary and Change Summary information contained in a WebSphere Process Server DataObject that is converted into a BusObj is lost. Applications generated by migration from WebSphere InterChange Server or WebSphere Business Integration Server Express will use the WebSphere InterChange Server or WebSphere Business Integration Server Express APIs in WebSphere Process Server, so Event Summary and Change Summary cannot be used with these applications until the code is manually updated to stop using any of the WebSphere InterChange Server or WebSphere Business Integration Server Express APIs. **Solution:** Remove all use of WebSphere InterChange Server or WebSphere Business Integration Server Express APIs or change them to WebSphere Process Server APIs.

Related concepts

"Troubleshooting migration from WebSphere InterChange Server or WebSphere Business Integration Server Express"
Find solutions to problems you encounter with migration as well as instructions for turning on logging and tracing.

 Compensation handling in business processes
Compensation processing is a means of handling faults in a running process instance for which compensation is defined in the process model. Compensation reverses the effects of operations, which were committed up to when the fault occurred, to get back to a consistent state.

Related reference

"Postmigration considerations" on page 141
When applications have been migrated from WebSphere InterChange Server or WebSphere Business Integration Server Express to WebSphere Process Server, special attention is required in some areas to enable migrated applications to function in WebSphere Process Server consistently with their intended function due to differences between the architectures of WebSphere Process Server and WebSphere InterChange Server or WebSphere Business Integration Server Express.

Troubleshooting migration from WebSphere InterChange Server or WebSphere Business Integration Server Express

Find solutions to problems you encounter with migration as well as instructions for turning on logging and tracing.

Related concepts

"Limitations when migrating from WebSphere InterChange Server or WebSphere Business Integration Server Express" on page 177
Some characteristics of WebSphere InterChange Server or WebSphere Business Integration Server Express are not precisely duplicated by WebSphere Process Server. Therefore you might have to modify your applications after migration to get them to perform as they did in WebSphere InterChange Server or WebSphere Business Integration Server Express.

Related reference

"Postmigration considerations" on page 141
When applications have been migrated from WebSphere InterChange Server or WebSphere Business Integration Server Express to WebSphere Process Server, special attention is required in some areas to enable migrated applications to function in WebSphere Process Server consistently with their intended function due to differences between the architectures of WebSphere Process Server and WebSphere InterChange Server or WebSphere Business Integration Server Express.

"Premigration considerations" on page 132
Consider these guidelines for the development of integration artifacts for WebSphere InterChange Server or WebSphere Business Integration Server Express to ease the migration of WebSphere InterChange Server or WebSphere Business Integration Server Express artifacts to WebSphere Process Server.

Enabling logging and tracing for supported WebSphere InterChange Server or WebSphere Business Integration Server Express APIs

Enable logging and tracing for the supported WebSphere InterChange Server or WebSphere Business Integration Server Express APIs through the administrative console.

About this task

If your migrated application includes any supported WebSphere InterChange Server or WebSphere Business Integration Server Express APIs, you can enable logging and tracing for them for troubleshooting purposes.

Procedure

1. Launch the administrative console.
2. From the left (navigation) panel, select **Troubleshooting > Logs and Trace**.
3. In the right panel, select the name of the server on which you want to enable logging and tracing.
4. In the right panel, under "General properties," select **Change Log Level Details**.
5. Select the Runtime tab. (Selecting the Runtime tab allows you to make this change in real time without requiring you to restart the server.)
6. Add the name of the package followed by `=all` to the list of logged packages in the box on the screen. Separate this new entry from any existing entries with a colon. For example, `CxCommon=all`. In this case, `CxCommon` is the name of the package for a set of supported WebSphere InterChange Server or WebSphere Business Integration Server Express APIs. Specifying `all` enables all logging and tracing. See Supported WebSphere InterChange Server APIs for a list of the APIs, including their package names.
7. Select **Apply**.
8. To keep this configuration after the server is restarted, select the **Save runtime changes to configuration as well** check box.
9. Select **OK**.
10. When the next screen appears, select **Save** to save your changes.

Related information

Supported WebSphere InterChange Server APIs

Failure trying to serialize an object that is not serializable in a migrated BPEL file

If a serialization failure occurs in a BPEL file generated by the migration, you might be able to modify it to prevent the failure from occurring.

Problem: A serialization failure occurs in a custom snippet node of a business process execution language (BPEL) file generated by migration because an attempt is made to serialize an object that is not serializable.

Cause: In WebSphere InterChange Server or WebSphere Business Integration Server Express, a Collaboration Template is compiled into a single Java class. In WebSphere Process Server, each node in a BPEL file might compile into a separate Java class. In WebSphere InterChange Server or WebSphere Business Integration Server Express, a variable can be declared once and shared throughout the various steps of a Collaboration Template. To simulate that behavior in the migrated BPEL file, each variable used in a code snippet must be retrieved at the start of the snippet and saved at the end of the snippet. Variables defined in WebSphere InterChange Server or WebSphere Business Integration Server Express Port definitions become BPEL variables. These are retrieved into BusObj variables at the beginning of each snippet (if referenced in the snippet) and saved back to the BPEL variables at the end of each snippet. For example, a retrieval at the beginning of snippets looks like this:

```
BusObj tempBusObj = null;if (tempBusObj_var != null) { tempBusObj =
      new BusObj(tempBusObj_var); };
```

and a save at the end of snippets looks like this:

```
if (tempBusObj == null) { tempBusObj_var = null; } else { tempBusObj_var =
      tempBusObj.getBusinessGraph(); }
```

Other variables used in the WebSphere InterChange Server or WebSphere Business Integration Server Express snippet code are serialized and stored as a String in a BPEL variable named *CollabTemplateName_var*. These variables are deserialized at the beginning of each BPEL snippet, and then serialized and saved at the end of each BPEL Snippet that they are referenced in. For example, objects are retrieved like this:

```
BusObj tempBusObj = (BusObj)BaseCollaboration.deserialize
  (FrontEndCollab_var.getString("tempBusObj"));
```

and objects are saved like this:

```
FrontEndCollab_var.setString("tempBusObj", BaseCollaboration.serialize(tempBusObj));
```

If the type of the object being serialized is not serializable, then using serialize and deserialize will fail when the BPEL is run.

Solution: After migration, modify the BPEL file as follows:

- For any variable that is not Java-serializable, update the BPEL snippets to remove the serialization and deserialization statements. If the variable needs to be shared across snippets (instead of being re-created in each snippet) another method must be used to preserve the value of the variable across snippets.
- Manually define BPEL variables for variables of type BusObj that are not declared in the WebSphere InterChange Server or WebSphere Business Integration Server Express Port definitions but are used on Partner Invokes. This is a manual step because variables used during invokes in WebSphere Process Server must be strongly typed, and the migration tools cannot accurately determine that type from the WebSphere InterChange Server or WebSphere Business Integration Server Express snippets.

  Note: The naming convention used by the migration tools is to add `_var` to the name of the variable in the snippet code when naming the BPEL variables. For example, for a variable called tempBusObj in the snippet code, the migration tools will create a BPEL variable named `tempBusObj_var`.
- For variables that must be declared manually as BPEL variables, change the BPEL snippet code so that it uses the "deserialize/serialize" method of preserving these variables rather than the "retrieve from/store into BPEL variable" method of preserving these variables.

New behavior for heritage APIs in WebSphere Process Server, version 7.0

In version 7.0 of WebSphere Process Server, heritage APIs use WebSphere Process Server Service Data Objects to store attribute states and data that were formerly stored by the BusinessObjectInterface interface. As a result, the behavior of some method calls in the BusinessObjectInterface and CxObjectContainerInterface interfaces have changed.

The major change to the heritage APIs (HAPIs) in WebSphere Process Server version 7.0 is that the WebSphere InterChange Server BusinessObjectInterface

interface is no longer the root storage object for HAPI. Instead, a WebSphere Process Server Service Data Object (SDO) is now used to store attribute states and data.

If you use the Java equivalence operator and weakly typed attribute principles, the behavior of the method calls in the BusinessObjectInterface and CxObjectContainerInterface interfaces is different, as described in the following sections:

- "Using the Java equivalence operator when performing a Set operation followed by a Get operation"
- "Using the Java equivalence operator when setting a BusinessObjectInterface object to more than one target attribute" on page 183
- "Using the Java equivalence operator when you set and retrieve a BusinessObjectInterface object from the CxObjectContainerInterface interface" on page 184
- "Using weakly typed attribute data types for the BusObj class validData methods" on page 185

Using the Java equivalence operator when performing a Set operation followed by a Get operation

A different BusinessObjectInterface object is returned when performing a Set operation followed by a Get operation of a BusinessObjectInterface object for one target attribute. The following table describes the previous behavior, current behavior, and an example of what to change if you previously had used the Java equivalence operator when performing a Set operations followed by a Get operation.

Table 10. Behavior changes: Using the Java equivalence operator with Set and Get operations

Behavior type	Description
Behavior prior to WebSphere Process Server version 7.0	The same BusinessObjectInterface container that was set was also retrieved, and you could use the Java equivalence operator "==" to determine if they were the same. Example: `boolean b = (JavaObjectA == JavaObjectB)`
Behavior after WebSphere Process Server version 7.0	The original BusinessObjectInterface container is discarded, and when you perform a Get operation to retrieve the BusinessObjectInterface object, a new container is created. The returned container is not the same object, but the root object that it wraps is the same object. A new method, isEquivalent, has been added to the BusinessObjectInterface class: BusinessObjectInterface.isEquivalent(BOI). When you want to determine if the two BusinessObjectInterface objects are equivalent, use the isEquivalent method to perform the comparison.

Table 10. Behavior changes: Using the Java equivalence operator with Set and Get operations (continued)

Behavior type	Description

Example of new behavior	The following example shows the use of isEquivalent. You have a BusinessObjectInterface object of type MasterBusinessObject with the attribute Attr_Nine, which is a BusinessObjectInterface object of type HelloWorld:
	`BusinessObjectInterface mboBOI, hw1BOI, hw2BOI;`
	`hw1BOI.setAttrValue("Message", "hw1BOI_message");`
	`hw1BOI.setVerb("Create");`
	`mboBOI.setAttrValue("Attr_Nine", hw1BOI);`
	`hw2BOI = mboBOI.getAttrValue("Attr_Nine");`
	
	Instead of:
	
	`boolean result = (hw1BOI == hw2BOI); assertTrue(result);`
	
	Use this:
	
	`boolean result = hw1BOI.isEquivalent(hw2BOI); assertTrue(result);`

Using the Java equivalence operator when setting a BusinessObjectInterface object to more than one target attribute

Setting a BusinessObjectInterface object to more than one target attribute sets a cloned object. This applies both to elements of a BusObjArray class and to multiple target attributes. The following table describes the previous behavior, current behavior, and an example of what to change if you previously had used the Java equivalence operator when setting a BusinessObjectInterface object to more than one target attribute.

Table 11. Behavior changes: Using the Java equivalence operator with more than one target attribute

Behavior type	Description
Behavior prior to WebSphere Process Server version 7.0	You could set a BusinessObjectInterface object to multiple locations, and all of the locations contained a reference to the original BusinessObjectInterface object. If you changed the attribute in one BusinessObjectInterface object, that change was reflected in all of the other references for that object.
Behavior after WebSphere Process Server version 7.0	Service Data Object (SDO) rules prevent you from setting the same SDO to more than one target property. If you try to set the SDO to more than one target property, the SDO moves from one attribute to the next, leaving a "null" value at the previous attribute location. Now, instead of leaving a "null" value when the BusinessObjectInterface object is set to a second, third, and so on, location, the object is cloned into the multiple locations. For example, you have a BusinessObjectInterface object of the type MasterBusinessObject, with the attributes Attr_Nine and Attr_Eleven that are of the type HelloWorld. If you set the same HelloWorld object to both attributes, then Attr_Nine is assigned the original object, and Attr_Eleven is assigned a clone. The clone is a snapshot of the object at the time that it is cloned. If you want to determine if two BusinessObjectInterface objects are equivalent, do not use the Java equivalence operator; instead, use the isEquivalent method to perform the comparison.

Table 11. Behavior changes: Using the Java equivalence operator with more than one target attribute (continued)

Behavior type	Description

Example of new behavior	The following example shows the use of isEquivalent and clones. You have a BusinessObjectInterface object of type MasterBusinessObject, with the attributes Attr_Nine and Attr_Eleven that are of type HelloWorld:
	`BusinessObjectInterface mboBOI;`
	`BusinessObjectInterface hw1BOI, hw2BOI, hw3BOI;`
	`hw1BOI.setAttrValue("Message", "hw1BOI_message");`
	`hw1BOI.setVerb("Create");`
	`mboBOI.setAttrValue("Attr_Nine", hw1BOI);`
	`mboBOI.setAttrValue("Attr_Eleven", hw1BOI);`
	`hw2BOI = mboBOI.getAttrValue("Attr_Nine");`
	`hw3BOI = mboBOI.getAttrValue("Attr_ Eleven ");`
	
	Instead of:
	
	`boolean result = hw2BOI == hw3BOI); assertTrue(result);`
	
	Use isEquivalent instead:
	
	`boolean result = hw2BOI.isEquivalent(hw3BOI); assertTrue(result);`
	
	The cloned objects do not share a reference, and changes to the original BusinessObjectInterface object are not reflected in the cloned BusinessObjectInterface object:
	`hw1BOI.setAttrValue("Message", "hw1BOI_message changed");`
	`boolean result = hw1BOI.isEquivalent(hw2BOI);`
	`assertTrue(result);`
	`boolean result = hw1BOI.isEquivalent(hw3BOI);`
	`assertFalse(result);`
	`boolean result = hw2BOI.isEquivalent(hw3BOI);`
	`assertFalse(result);`

Using the Java equivalence operator when you set and retrieve a BusinessObjectInterface object from the CxObjectContainerInterface interface

The following table describes the previous behavior, current behavior, and an example of what to change if you previously had used the Java equivalence operator when setting and retrieving a BusinessObjectInterface object from the CxObjectContainerInterface interface.

Table 12. Behavior changes: Using the Java equivalence operator with the CxObjectContainerInterface interface

Behavior type	Description
Behavior prior to WebSphere Process Server version 7.0	When you set and then retrieved a BusinessObjectInterface object from the CxObjectContainerInterface interface, you could use the Java equivalence operator "==" because the BusinessObjectInterface container that was retrieved was the same BusinessObjectInterface container that was set.
Behavior after WebSphere Process Server version 7.0	You must use the BusinessObjectInterface.isEquivalent(BOI) method.

Table 12. Behavior changes: Using the Java equivalence operator with the CxObjectContainerInterface interface (continued)

Behavior type	Description

Example of new behavior	The following JUnit test code demonstrates the old and new behavior.
	`CxObjectContainerInterface testCxObjectContainerInt;`
	`BusinessObjectInterface mB01, mB02, mB03;`
	`testCxObjectContainerInt.insertBusinessObject(mB01);`
	`testCxObjectContainerInt.setBusinessObject(1, mB01);`
	`BusinessObjectInterface mB02 = testCxObjectContainerInt.`
	`getBusinessObject(0));`
	`BusinessObjectInterface mB03 = testCxObjectContainerInt.`
	`getBusinessObject(1));`
	`assertTrue(mB01 == mB02);`
	`assertTrue(mB01 == mB03);`
	`assertTrue(mB02 == mB03);`
	
	This Java equivalence operator no longer works because the BusinessObjectInterface object that is returned by CxObjectContainerInterface.getBusinessObject(int index) is not the same Java object that was set to CxObjectContainerInterface.
	
	In the following code, the equivalence operator is replaced by the method BusinessObjectInterface.isEquivalent(BOI):
	`boolean result1 = mB01.isEquivalent(mB02)`
	`assertTrue(result1);`
	`boolean result2 = mB01.isEquivalent(mB03)`
	`assertFalse(result2);`
	`boolean result3 = mB02.isEquivalent(mB03)`
	`assertFalse(result3);`
	
	The cloned objects do not share a reference, and changes to the original BusinessObjectInterface object are not reflected in the cloned BusinessObjectInterface object:
	`hw1BOI.setAttrValue("Message", "hw1BOI_message changed");`
	`boolean result = mB01.isEquivalent(mB02;`
	`assertTrue(result);`
	`boolean result = mB01.isEquivalent(mB02;`
	`assertFalse(result);`
	`boolean result = mB02.isEquivalent(hw3BOI);`
	`assertFalse(result);`

Using weakly typed attribute data types for the BusObj class validData methods

The following table describes the previous behavior, current behavior, and an example of what to change if you previously had used weakly typed attribute data types in WebSphere InterChange Server or WebSphere Business Integration Server Express for the BusObj class validData methods.

Table 13. Behavior changes: Using weakly typed attribute data types for BusObj class validData methods

Behavior type	Description
Behavior prior to WebSphere Process Server version 7.0	For the BusObj class validData methods, attribute data types were weakly typed in WebSphere InterChange Server or WebSphere Business Integration Server Express. This allowed some odd data-type combinations. For example, if a business object had an attribute that was of type boolean but you used a set method that had a string parameter, you were able to set the string "not a boolean" into an attribute that was type boolean. As long as you used the getString method, they could get the string "not a boolean" back.
Behavior after WebSphere Process Server version 7.0	These attribute data types are now strongly typed. If a data type was valid but now is not valid, a CollaborationException exception is thrown, with message number 1802. Because WebSphere Process Server is strongly typed, you cannot put a String value into an Attribute of type boolean. Even if you used the Java conversions for strings into the boolean values of *true* and *false*, there is no way to return the original value of "not a boolean". The only possible returned value is *true* or *false*. Therefore, attributes are now strongly typed for doubles-floats or int-long; these can be used interchangeably where Java provides automatic casting. However, as with any casting of types, you can expect some loss of precision when fields are demoted. If a type is not valid for the attribute to which it is set, but it used to be valid in WebSphere InterChange Server or WebSphere Business Integration Server Express, then a CollaborationException exception is thrown, with message number 1802. This is a new message number; the message definition is located in the InterchangeSystem.txt message file.

Example of new behavior	If a type is not valid for the Attribute to which it is set, but it used to be valid in WebSphere InterChange Server, a CollaborationException with message number 1802 will be thrown. This is a new message number, the Message definition is located in the InterchangeSystem.txt message file:
	`try`
	`{`
	`BusObj mBO = new BusObj("MasterBusinessObject");`
	`mBO.set("Attr_Two", "xxx");`
	`fail("Expected CollaborationException not thrown");`
	`}`
	`catch (CollaborationException e)`
	`{`
	`int a = e.getMsgNumber();`
	`String b = e.getSubType();`
	`String c = e.getMessage();`
	`String d = e.toString();`
	`assertEquals("exception_msgNumber", 1802, a);`
	`assertEquals("exception_type", "AttributeException", b);`
	`assertEquals("exception_message",`
	`"Error 1802 The attribute \"Attr_Two\" in SDO MasterBusinessObject is of type boolean and is not allowed to be set with a value \"xxx\" of type String. Error1802", c);`
	
	`assertEquals("exception_toString", "AttributeException: Error 1802 The attribute \"Attr_Two\" in SDO MasterBusinessObject is of type boolean and is not allowed to be set with a value \"xxx\" of type String. Error1802", d);}`

A microflow is not compensated

A microflow has called a service, and the process fails, but the undo service is not called.

Resolution

There are various conditions that must be met to trigger the compensation of a microflow. Check the following:

1. Log on to the Business Process Choreographer Explorer and click **Failed Compensations** to check whether the compensation service has failed and needs to be repaired.
2. The compensation of a microflow is triggered only when the transaction for the microflow is rolled back. Check whether this is the case.
3. The compensationSphere attribute of the microflow must be set to required.
4. A compensation service is run only if the corresponding forward service has not participated in the microflow's transaction. Ensure that the forward service does not participate in the navigation transaction, for example, on the reference of the process component, set the Service Component Architecture (SCA) qualifier suspendTransaction to True.

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