

The IBM logo is centered on a white background. It consists of the letters 'IBM' in a bold, blue, sans-serif font. Each letter is composed of eight horizontal stripes, with the top and bottom stripes being slightly thicker than the others. The logo is positioned in the lower-middle section of the page, which is framed by a large, light blue circular graphic with a gradient and a subtle shadow effect.

Contents

- Migrating..... 1**
 - Preparing for migration..... 1
 - Supported migration paths..... 2
 - Migration planning..... 2
 - Behavioral changes in Version 11.0..... 4
 - IBM MQ and migration..... 9
 - IBM App Connect Enterprise migration options..... 10
 - Example migration scenarios..... 11
 - Performing pre-migration tasks..... 22
 - Backing up resources before migration..... 23
 - Updating ODBC definitions..... 23
 - Running the Transformation Advisor Tool..... 25
 - Rules for the Transformation Advisor tool..... 29
 - Migrating to IBM App Connect Enterprise Version 11.0..... 40
 - Performing extract migration of an integration node or integration server..... 40
 - Performing parallel migration for an integration node 44
 - Migrating development resources..... 45
 - Performing post-migration tasks..... 48
 - Reviewing technical changes in IBM App Connect Enterprise Version 11.0..... 48
 - Setting up a command environment..... 49
 - Reconfiguring administration security..... 50
 - Setting up a global cache..... 50
 - Migrating deployable resources..... 51

- Index..... 57**

Migrating

To migrate an integration server or integration node from IBM® Integration Bus Version 10.0 or Version 9.0 to IBM App Connect Enterprise Version 11.0, plan your migration strategy, perform pre-migration tasks, migrate your domain components, or server configuration and resources, then complete post-migration tasks.

Before you begin

Tip: Check for migration information updates on the [IBM App Connect Enterprise support web page](#).

Preparing for migration

Plan the order and extent of the migration of components and resources to IBM App Connect Enterprise Version 11.0.

About this task

To prepare for migration, complete the following steps.

Procedure

1. Check that your current installation of IBM Integration Bus Version 10.0 or IBM Integration Bus Version 9.0 is at a supported level for migration.
Details are provided in [“Supported migration paths”](#) on page 2.
For the latest details of all supported levels of hardware and software, see the [IBM App Connect Enterprise system requirements website](#).
2. Some function that was available in IBM Integration Bus is not currently available in IBM App Connect Enterprise Version 11.0; elements of this function are to be made available in fix pack updates. For details of the unavailability of commands, policy types, and other resources in IBM App Connect Enterprise at this fix pack release, see the [IBM App Connect Enterprise FAQ website](#).
3. Determine your migration priorities; see [“Migration planning”](#) on page 2.
4. Understand the options that are available to install IBM App Connect Enterprise Version 11.0 components on the same computer as components from IBM Integration Bus.
For more information about the options that are available for versions of IBM App Connect Enterprise and IBM Integration Bus to coexist, see [Coexistence of with previous versions](#).
5. Learn about new and changed function in Version 11.0 by reading [What's new in ?](#).
These changes might affect how you want to use your migrated components in the future.
6. Understand the IBM MQ configuration options that you can include in your migration plan; see [“IBM MQ and migration to IBM App Connect Enterprise Version 11.0”](#) on page 9.
These changes might affect your choice of integration node migration option.
7. Review the options for migrating your integration nodes and integration servers and decide whether to use extract migration or parallel migration; see [“IBM App Connect Enterprise migration options”](#) on page 10
8. Review the technical changes in behavior in Version 11.0; see [“Behavioral changes in Version 11.0”](#) on page 4.
These changes might affect your post-migration development tasks.
9. Check the requirements for other products on which Version 11.0 components might depend.
If you configured your message flows to use external resources, such as databases, or event monitoring applications, you might have to modify your configuration. You can find details of supported

versions of complementary products on the [IBM App Connect Enterprise system requirements](#) web page.

What to do next

After you plan your migration, complete the pre-migration tasks by following the instructions in [“Performing pre-migration tasks”](#) on page 22.

Supported migration paths

You can migrate to IBM App Connect Enterprise Version 11.0 from IBM Integration Bus Version 10.0 and Version 9.0.

Note: For the latest details of all supported levels of hardware and software, see the [IBM App Connect Enterprise system requirements](#) website.

You can either perform parallel migration to migrate integration nodes, or extract migration to migrate integration servers to IBM App Connect Enterprise Version 11.0.

You can migrate from IBM Integration Bus Version 10.0 or Version 9.0 only to either a full edition or Standard edition of IBM App Connect Enterprise Version 11.0. Do not migrate integration nodes from an earlier version to the Developer Edition.

Migration planning

The order in which you migrate your IBM Integration Bus Version 10.0 or Version 9.0 environment depends on whether your priority is to include new development features, take advantage of new operational features, or simply implement a fully supported version of IBM App Connect Enterprise.

Your existing IBM Integration Bus environment might consist of a number of components:

- Integration nodes that support production applications
- Build systems that create deployable resources from the development source files
- Integration nodes that are used for testing applications
- Integration nodes that are used for developing applications
- Instances of the IBM Integration Toolkit

The order in which you migrate your environment to IBM App Connect Enterprise Version 11.0 is likely to depend on which of the following factors is most important to you:

- [“Supported version or new operational features”](#) on page 2
- [“New application features”](#) on page 3

In both circumstances, you can use parallel migration. If you are migrating an integration server, you can use the **mqsextractcomponents** command to migrate the configuration and resources that exist for your integration server to IBM App Connect Enterprise Version 11.0.

Supported version or new operational features

If your priority in migrating to IBM App Connect Enterprise Version 11.0 is simply to have an environment that is at a fully-supported version of IBM App Connect Enterprise, and you do not need to use any of the new Version 11.0 features immediately, there are a minimum number of steps you must complete.

If your priority in migrating to IBM App Connect Enterprise Version 11.0 is to use the new Version 11.0 operational features, you can update your integration nodes first. You can use existing development environments and application build processes, and deploy your existing BAR files until you are ready to migrate your development resources.

In either scenario, you migrate the components of your environment in the following order:

1. Migrate the integration nodes that support your test environment.

2. Implement new Version 11.0 operational functionality on your test environment or, at a minimum, update existing operational functionality:
 - If you are using IBM Integration Explorer in your existing environment, define new operational procedures that use the web user interface. You cannot use previous versions of IBM Integration Explorer to administer IBM App Connect Enterprise Version 11.0.
 - If you are using scripts to administer your existing environment, update any scripts that use commands that connect to integration nodes. The parameters that are used by commands that connect to integration nodes have changed in Version 11.0.
3. Migrate the integration nodes that support your production environment.
4. Implement new Version 11.0 operational functionality on your production environment or, at a minimum, update existing operational functionality.
5. If you have any integration nodes that support your development environment, migrate these integration nodes to Version 11.0.
6. Update your build system to create Version 11.0 deployable resources. If required, update build scripts to take advantage of new Version 11.0 operational functionality but, at a minimum, update any scripts that use commands that connect to integration nodes.
7. Install IBM App Connect Enterprise Version 11.0 on your developer workstations. If you cannot migrate all developer workstations at the same time, you must create separate development streams. You cannot use Version 11.0 development tools to build applications for an environment that is running a previous version of IBM App Connect Enterprise.
8. Import development resources from your previous IBM Integration Toolkit.

For the steps that tell you to migrate integration nodes, use parallel migration; see [“IBM App Connect Enterprise migration options”](#) on page 10.

When you have imported all the development resources, you can uninstall the previous versions of IBM Integration Toolkit, and any integration nodes that you do not want to migrate.

New application features

If your priority in migrating to IBM App Connect Enterprise Version 11.0 is to develop applications that take advantage of new functions and features in Version 11.0, you can install a new development environment alongside your existing development environment, and create new build, test, and production environments to support your Version 11.0 development.

In this scenario, you migrate the components of your environment in the following order:

1. Install IBM App Connect Enterprise Version 11.0 on your developer workstations. To maintain existing applications in your existing environment while you are building new applications for Version 11.0, you must run two development streams. You cannot use Version 11.0 development tools to build applications for an environment that is running a previous version of IBM App Connect Enterprise.
2. If you are updating an existing application, import development resources from your previous IBM Integration Toolkit.
3. Develop applications that take advantage of the Version 11.0 features.
4. Create a new build system that creates Version 11.0 deployable resources. You can use build scripts from your previous version but you must update any scripts that use commands that connect to integration nodes. The parameters that are used by commands that connect to integration nodes have changed in Version 11.0.
5. Create one or more IBM App Connect Enterprise Version 11.0 integration nodes to support the testing of the Version 11.0 applications.
6. Update existing operational functionality:
 - If you are using IBM Integration Explorer in your existing environment, define new operational procedures that use the web user interface. You cannot use previous versions of IBM Integration Explorer to administer IBM App Connect Enterprise Version 11.0.

- If you are using scripts to administer your existing environment, update any scripts that use commands that connect to integration nodes.
- 7. Deploy Version 11.0 applications to the Version 11.0 testing environment as required.
- 8. Create one or more IBM App Connect Enterprise Version 11.0 integration nodes to support production use of the Version 11.0 applications.
- 9. Deploy Version 11.0 applications to the Version 11.0 production environment as required.
- 10. Migrate or deprecate applications from the original environment as required.

This type of migration is known as parallel migration; see [“IBM App Connect Enterprise migration options” on page 10](#).

When all applications are migrated to Version 11.0, you can uninstall the original environment.

Behavioral changes in Version 11.0

Depending on your current version of IBM Integration Bus, IBM App Connect Enterprise Version 11.0 might introduce several technical changes in behavior. These changes might affect your post-migration development tasks.

The changes that affect you vary depending on your current version. Review the following changes to see how your post-migration development tasks might be affected:

- When you are migrating from IBM Integration Bus Version 10.0, see the following topics:
 - [“Platform support” on page 5](#)
 - [“Policies control and update message flow and message flow node properties” on page 5](#)
 - [“Nodes that are no longer available” on page 5](#)
 - [“IBM App Connect Enterprise Version 11.0 REST administration API \(REST API\) Version 2 is available to issue administration commands” on page 5](#)
 - [“Additional options for administration security” on page 6](#)
 - [“Administration scripts might need to be updated” on page 6](#)
 - [“Publication of statistics to the web user interface is enabled by default” on page 7](#)
 - [“UUIDs are not assigned to the Integration Node, Integration Server, Application, or Message Flow components” on page 7](#)
 - [“Message flow monitoring” on page 7](#)
 - [“LDAP authentication and local passwords” on page 8](#).
- If you are migrating from IBM Integration Bus Version 9.0, see the following topics:
 - [“Platform support” on page 5](#)
 - [“Policies control and update message flow and message flow node properties” on page 5](#)
 - [“Nodes that are no longer available” on page 5](#)
 - [“IBM App Connect Enterprise Version 11.0 REST administration API \(REST API\) Version 2 is available to issue administration commands” on page 5](#)
 - [“Web user interface extended to support administration tasks” on page 6](#)
 - [“Additional options for administration security” on page 6](#)
 - [“Administration scripts might need to be updated” on page 6](#)
 - [“IBM MQ and migration to IBM App Connect Enterprise Version 11.0” on page 9](#)
 - [“Enhanced flexibility in interactions with IBM MQ” on page 8](#)
 - [“Message maps: change in behavior” on page 6](#)
 - [“Message maps are validated at deployment time” on page 7](#)
 - [“IBM App Connect Enterprise Toolkit help topics might include details of function that is not available to you” on page 6](#)

- [“Message flow monitoring” on page 7](#)

Platform support

IBM App Connect Enterprise Version 11.0 is available on:

- Windows
- Linux[®]
- AIX[®]

For detailed information about the operating systems that are supported, visit the [IBM App Connect Enterprise Version 11.0 system requirements web page](#).

The Statement of general direction in the [IBM App Connect Enterprise Version 11.0 announcement letter](#) outlines intention for support for other operating systems.

Policies control and update message flow and message flow node properties

At earlier releases, configurable services were used to control and update connection properties and other operational properties of message flows and message flow nodes at run time. In IBM App Connect Enterprise Version 11.0, you can use policies for these tasks.

You can migrate your existing configurable policies by using the **mqsextractcomponents** command. If you are performing parallel migration, you need to create policies to replace your configurable services.

For more information, see [Overriding properties at run time with policies](#).

Nodes that are no longer available

Your message flows from earlier versions work at IBM App Connect Enterprise Version 11.0. However, a number of message flow nodes are not available at Version 11.0; if your earlier message flows include them, consider reworking them.

The following nodes are not available in IBM App Connect Enterprise Version 11.0:

- DataDelete node
- DataInsert node
- DataUpdate node
- Extract node.
- Mapping node from WebSphere[®] Message Broker Version 7.0 and earlier versions
- MQOptimizedFlow node
- Real-timeOptimizedFlow node
- Warehouse node

If you try to use a message flow that contains any of these nodes, the message flow does not start and message BIP2355 is written to the syslog. The message mapping (.msgmap) on these Version 7.0 nodes can be viewed but they cannot be deployed to the runtime in IBM App Connect Enterprise Version 11.0.

For more information about other nodes that are not available in IBM App Connect Enterprise Version 11.0, see [Built-in nodes](#).

IBM App Connect Enterprise Version 11.0 REST administration API (REST API) Version 2 is available to issue administration commands

IBM App Connect Enterprise Version 11.0 does not include the IBM Integration API. Instead, you can use the REST API (Version 2) that is provided in IBM App Connect Enterprise Version 11.0 to issue administration commands.

For more information, see [Managing resources by using the administration REST API](#).

Web user interface extended to support administration tasks

IBM App Connect Enterprise Version 11.0 does not include IBM Integration Explorer. Instead, the web user interface is extended to support integration node administration tasks such as creating and managing integration servers, and deploying and managing resources.

For more information, see [web user interface](#).

Additional options for administration security

In IBM App Connect Enterprise Version 11.0, you have two options for configuring administration security. As in previous versions, you can configure queue-based permissions by using IBM MQ queues on the queue manager that is specified on the integration node. Alternatively, you can configure file-based permissions on your integration node, which you set by using the `mqsichangefileauth` command.

If you configure administration security and then you perform parallel migration to integration nodes that are not configured with queue managers, you must reconfigure your administration security to use file-based permissions. Any security permissions that are set before migration are not retained after migration.

For more information, see [Authorizing users for administration](#).

Administration scripts might need to be updated

A number of commands are not available at this release. In addition, some parameters for administration commands that connect to integration nodes and integration servers were changed. For more information, see [Commands](#). The `MQBrokerConnectionParameters` IBM Integration API class is also deprecated; see the Javadoc for the [IBM Integration API](#).

IBM App Connect Enterprise Toolkit help topics might include details of function that is not available to you

By default, the IBM App Connect Enterprise Toolkit is now configured to use online product documentation to provide context-sensitive help from the latest information that is available. In previous versions, the IBM App Connect Enterprise Toolkit contained local help documentation, which was not automatically updated when new information became available. There is no difference in the way that help is displayed in the IBM App Connect Enterprise Toolkit. However, the content of the IBM App Connect Enterprise section of the online product documentation is updated when fix packs are made available. If you do not deploy the latest fix pack, the help information might include details of function that is not available to you.

If you do not want to access the online product documentation, you can download and install a local source of product documentation. For more information about local documentation options, see [Adding documentation to the IBM App Connect Enterprise Toolkit](#).

Message maps: change in behavior

In IBM App Connect Enterprise Version 11.0, the behavior of the **Assign** transform in the Graphical Data Mapping editor has been corrected when you assign an empty string to an element.

- In previous versions, when you perform an **Assign** transform on a target element that is defined as `xsd:string`, the Graphical Data Mapping editor sets the internal NULL value in the message tree for that element.
- In IBM App Connect Enterprise Version 11.0, when you perform an **Assign** transform on a target element that is defined as `xsd:string`, the Graphical Data Mapping editor sets the element value to the empty string `' '` value.

For message formats such as XML, this change in behavior has no impact on the message flow. However, if your message flow has logic that tests that the value of an element is the internal NULL value, you must

modify the test to look for the empty string value. Alternatively, you must modify the map to use the new **iib:nullvalue()** function. To call the new function, you use a **Custom XPath** transform.

Message maps are validated at deployment time

In IBM App Connect Enterprise Version 11.0, message maps are prepared for execution on deployment instead of when the first message flows through the Mapping node. For more information, see [Deploying message maps](#).

Java isolation is active and cannot be disabled

Versions 8 and 9 of IBM App Connect Enterprise did not support Java™ isolation in applications. All Java that was deployed to the execution group was loaded into a single class loader, which was used by all JavaCompute nodes in all applications. This behavior precluded the use of duplicate classes.

When you create an application in Version 10 of IBM App Connect Enterprise, Java isolation is enabled by default. For each application, a Java class loader is built that contains only the Java that is deployed in that application and any included static libraries.

When you create an application in Version 11 or later of IBM App Connect Enterprise, Java isolation is active and you cannot disable it.

Publication of statistics to the web user interface is enabled by default

When an integration node or integration server is created in IBM App Connect Enterprise Version 11.0.0.8 (or later), the publication of resource statistics and message flow snapshot statistics is activated by default. The publication of archive statistics is turned off by default. The **publicationOn** property in the `server.conf.yaml` or `node.conf.yaml` file is explicitly set to `active` and the **outputFormat** property is set to `json`, which enables the publication of snapshot data to the web user interface. The **reportingOn** property is explicitly set to `true`, which enables resource statistics for the integration node or server to be published automatically to the web user interface.

For integration nodes and servers that were created before IBM App Connect Enterprise Version 11.0.0.8, the publication of all statistics was turned off by default. To enable the publication of statistics for integration nodes or servers that were created before V11.0.0.8, edit the relevant `.conf.yaml` file and activate the **publicationOn** and **reportingOn** properties, as required.

For more information, see [Managing resource statistics collection](#) and [Configuring the collection of message flow statistics by using a .yaml configuration file](#).

UUIDs are not assigned to the Integration Node, Integration Server, Application, or Message Flow components

In IBM App Connect Enterprise Version 11.0, UUIDs are not assigned to the Integration Node, Integration Server, Application, or Message Flow components. When these fields are displayed, for example in Monitoring Events they are set to all 0's.

For more information, see [The monitoring event](#).

Message flow monitoring

Monitoring events are issued in a new format that provides additional information. However, the format that was used in WebSphere Message Broker Version 8.0.0.0, IBM Integration Bus Version 9.0, and IBM Integration Bus Version 10.0 can also be used in IBM App Connect Enterprise Version 11.0 by editing the `eventFormat` property in the `Monitoring.MessageFlow` section of the `.conf.yaml` file.

For more information, see [Activating monitoring](#).

LDAP authentication and local passwords

If a web user account has a local password, and LDAP authentication is enabled, the local password is ignored. When LDAP authentication is enabled, all web user logins must be authenticated by using LDAP. Any local passwords are ignored. For more information, see [Enabling LDAP authentication and command - , , and systems](#).

Enhanced flexibility in interactions with IBM MQ

Greater flexibility was introduced in IBM Integration Bus Version 10.0 in its interactions with IBM MQ. IBM App Connect Enterprise Version 11.0 maintains this enhanced flexibility.

You can configure local or client connections to IBM MQ, enabling your integration nodes to get messages from, or put messages to, queues on a local or remote queue manager. You can configure either a local or client connection between your integration node and your queue manager, depending on the configuration of your existing architecture. If your IBM MQ queue manager is running on the same machine as your integration node, you can specify a local connection to the queue manager. Alternatively, if the IBM MQ queue manager that you want to connect to is hosted on a separate machine from IBM App Connect Enterprise, you can configure a client connection from your integration node so that it can access the messages on the remote queue manager.

When you configure a connection from an MQ node to an IBM MQ queue manager, you can optionally configure the connection to use a security identity for authentication, SSL for confidentiality, or both. The security identity, which passes user name and password security credentials to the queue manager, can be used on connections to local or remote queue managers. For connections to remote queue managers, you can choose whether to use the SSL protocol to provide confidentiality on the client connection. IBM App Connect Enterprise supports a subset of the SSL functionality that is supported by IBM MQ. For more information, see [Connecting to a secured queue manager](#).

Note: You cannot use a secured queue manager as the local default queue manager for an integration node or an integration server.

You can get messages from, or put messages to, IBM MQ queues on local or remote queue managers, by configuring the connection properties of the following MQ nodes:

- [node](#)
- [node](#)
- [node](#)
- [node](#)

Alternatively, you can specify a queue manager to be associated with the integration node by using the **-q** parameter on the **mqsicreatebroker** command. This queue manager is used by default for MQ processing in the message flow if no queue manager is specified explicitly on the MQ node. This queue manager is also used by some message flow nodes that require a queue manager to be specified on the integration node, such as the event-driven processing nodes used for aggregation, timeout, message collection, and message sequencing. The **-q** parameter specifies the name of a queue manager, but it does not create the queue manager automatically. You must define and start the queue manager as separate tasks, in addition to specifying it with the **mqsicreatebroker** command.

Message flows can contain multiple MQInput and MQOutput nodes, each of which can access different queue managers specified in the MQ node. For more information, see [node](#) and [node](#).

IBM MQ is not a prerequisite for using IBM App Connect Enterprise, which means that you can develop and deploy applications with IBM App Connect Enterprise independently of IBM MQ. However, some IBM App Connect Enterprise features require access to IBM MQ, including the MQ nodes and the event-driven processing nodes that are used for aggregation and timeout flows, message collections, and message sequences. IBM MQ is not provided as part of the IBM App Connect Enterprise installation package; however, when you purchase a license for IBM App Connect Enterprise, your license entitles you to install IBM MQ for use by App Connect Enterprise, within the terms of the license.

The following IBM App Connect Enterprise features require IBM MQ Server to be installed on the same machine as the integration node, and they are available for use only if you specify a queue manager on the integration node:

- Queue-based administration security (MQ is not required for file-based security)
- Global transactionality
- FTEInput and FTEOutput nodes
- CDInput and CDOOutput nodes
- Integration nodes with HTTP listeners
- HTTP proxy servlet
- High availability configurations

The integration node listener requires access to IBM MQ Server, so you must install it if you want to use an integration node listener to manage HTTP messages in your HTTP or SOAP flows. However, if you use HTTP nodes or SOAP nodes with the integration server embedded listener, they do not require access to IBM MQ.

The following IBM App Connect Enterprise features require access to system queues on a local queue manager (running on the same machine as the integration node) for the storage and retrieval of state information:

- Queue-based administration security (system queues are not required for *file-based* security).
- Integration node HTTP listener.
- HTTP proxy servlet.

The following IBM App Connect Enterprise features require access to system queues on either a local or remote queue manager for the storage and retrieval of state information:

- Record and replay.
- Event-driven processing nodes (aggregate, collector, sequence, resequence, and timeout nodes).

For information about creating the system queues, see [Creating the default system queues on an queue manager](#).

On Linux and AIX systems only, you must also configure the IBM MQ environment that you want the integration node to use before you start it. If you do not set the environment, your integration node might not run in the expected location. For more information, see [Setting the environment on and](#).

The MQInput, MQOutput, MQGet, and MQReply nodes require that IBM MQ is installed either locally or remotely, but they do not require a queue manager to be specified on the integration node unless you want to use this queue manager by default for your local MQ connection. For more information, see [Configuring a local connection to](#) and [Configuring a client connection to](#).

The SAPInput, SAPReply, and SAPRequest nodes require that either IBM MQ Client or Server is installed on the same machine as the integration node, and they require a queue manager to be specified on the integration node.

For a list of the main IBM App Connect Enterprise features, including information about the features that require the installation of IBM MQ Client or Server, see [features](#).

For a summary of the features that are new to IBM App Connect Enterprise Version 11.0, see [What's new in ?](#).

IBM MQ and migration to IBM App Connect Enterprise Version 11.0

Options for IBM MQ integration when you migrate to IBM App Connect Enterprise Version 11.0.

Greater flexibility was introduced in IBM Integration Bus Version 10.0 in its interactions with IBM MQ; IBM App Connect Enterprise Version 11.0 maintains this enhanced flexibility. See [“Enhanced flexibility in interactions with IBM MQ”](#) on page 8 and topologies.

As part of your migration process, you might want to configure your IBM App Connect Enterprise Version 11.0 deployment to take advantage of the greater flexibility.

In IBM Integration Bus Version 9.0, the integration layer of your architecture must contain IBM MQ (or IBM MQ) queue managers. If you have queues that you use in integration applications, you must have an existing IBM MQ (or IBM MQ) topology in which application messages are routed to the queue manager that is specified on the integration node by using IBM MQ (or IBM MQ) channels, remote queue definitions, and distributed messaging. It might be possible to simplify your system so that your message flows interact directly with remote queue managers, which might simplify the topology that you need to manage. This simplification requires that you redesign your message flows and your topology, and is more than just a migration of your existing solution. However, you might want to include these activities as part of your migration plans.

To determine which components of your existing deployment are using capabilities that require IBM MQ or (IBM MQ) queue managers to be a part of the integration layer of your IBM App Connect Enterprise Version 11.0 architecture, and which components are using capabilities that can integrate with IBM MQ application queues on a remote queue manager, see [Installing](#).

Depending on whether you can change your IBM MQ topology during the migration process, you have the following options for migrating integration nodes:

- Create a new Version 11.0 integration node and associate the integration node with a new queue manager. Add the new queue manager into your IBM MQ network, migrate the applications from the original integration node, and redirect application queues as required (parallel migration).
- Create a new Version 11.0 integration node and associate the integration node with the same queue manager as the original integration node. Migrate the applications from the original integration node, and keep the IBM MQ application endpoints and IBM MQ topology the same (parallel migration).
- If you use this option and either the original or the new integration node has message flows that contain the following message flow nodes, you must stop the original integration node before you start the new integration node. These message flow nodes use IBM MQ queues to save state, which cannot be shared between integration nodes:
 - AggregateControl
 - AggregateReply
 - AggregateRequest
 - Collector
 - Sequence
 - Resequence
 - TimeoutControl
 - TimeoutNotification
- Create a new Version 11.0 integration node and do not associate the integration node with a queue manager. Migrate the applications from the original integration node, and configure message flows that require IBM MQ queues to connect to specified queue managers, either by directly configuring the message flow or by using a policy (parallel migration).

For information about migration, see [“IBM App Connect Enterprise migration options”](#) on page 10.

IBM App Connect Enterprise migration options

IBM App Connect Enterprise Version 11.0 supports parallel migration and extract migration to migrate your application logic to Version 11.0 systems.

If you want to reproduce the integration node function on another computer, you can use *parallel migration* to associate the application logic on your existing integration node with a separate Version 11.0 integration node.

By using *extract migration*, you can migrate existing integration node and integration server configuration and resources to IBM App Connect Enterprise Version 11.0.

Restrictions

Consider the following restrictions before deciding whether to use parallel migration for your integration nodes.

IBM MQ enhancements

Greater flexibility was introduced in IBM Integration Bus Version 10.0 in its interactions with IBM MQ; IBM App Connect Enterprise Version 11.0 maintains this enhanced flexibility. Ensure that you are using a supported version of IBM MQ to take advantage of the greater flexibility.

If you use parallel migration, you can install and configure an IBM App Connect Enterprise Version 11.0 deployment that takes advantage of the IBM MQ flexibility, and then move your applications into the Version 11.0 deployment.

For more information, see [“IBM MQ and migration to IBM App Connect Enterprise Version 11.0” on page 9](#).

Maintaining administration security

If you have administration security configured in your current environment by using IBM MQ queues, and you perform parallel migration, you must recreate the equivalent administration security settings by using file-based permissions on your new integration nodes; see [Authorizing users for administration](#).

Example migration scenarios

Example migration scenarios demonstrate how to migrate message flows and applications from IBM Integration Bus Version 10.0 or IBM Integration Bus Version 9.0 to IBM App Connect Enterprise Version 11.0.

- [“Migrating an integration node with configurable services” on page 12](#)

This scenario shows how to migrate an IBM Integration Bus Version 10.0 or Version 9.0 integration node that has configurable services for external resources. The Version 10.0 or Version 9.0 node also has a specific user ID and password that is associated with one or more resources by using the **mqsisetdbparms** command.

- [“Migrating an integration node with multiple integration servers” on page 13](#)

This scenario uses the **mqsextractcomponents** command to migrate an IBM Integration Bus Version 10.0 or Version 9.0 integration node with multiple integration servers to IBM App Connect Enterprise Version 11.0 independent integration servers.

- [“Migrating an integration server that contains applications and a shared library” on page 15](#)

This scenario shows how to migrate an IBM Integration Bus Version 10.0 or Version 9.0 integration server with applications and a shared library to IBM App Connect Enterprise Version 11.0.

- [“Migrating an SAP integration flow” on page 16](#)

This scenario shows how to migrate an SAP integration flow that runs on IBM Integration Bus Version 10.0 or Version 9.0.

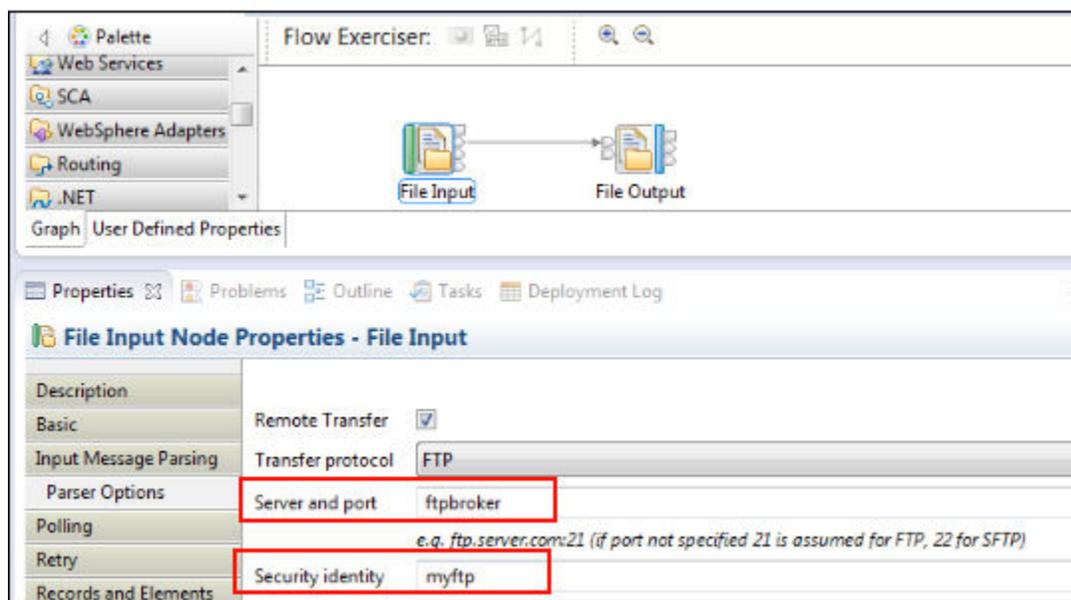
Migrating an integration node with configurable services

Migrate an IBM Integration Bus Version 10.0 or Version 9.0 integration node that has configurable services for external resources, and a specific user ID and password that is associated with one or more resources by using the **mqsisetdbparms** command.

About this task

In this scenario, a simple flow that contains a FileInput node and a FileOutput node is deployed to an IBM Integration Bus Version 10.0 or Version 9.0 integration server, then to an integration node. An FtpServer configurable server has been created by using the following command:

```
mqsicreateconfigurable-service MigrationNode -c FtpServer -o FtpServer01  
-n serverName,scanDelay,transferMode,connectionType,securityIdentity  
-v one.hursley.abc.com:123,20,Binary,ACTIVE,myftp
```



The following steps describe how to migrate the flow to IBM App Connect Enterprise Version 11.0.

Procedure

1. Back up the IBM Integration Bus Version 10.0 or Version 9.0 integration node by using the **mqsibackupbroker** command.

```
mqsibackupbroker IntegrationNode -d directory
```

A file with a name in the format `intNodeName_yyMMdd_HHmss.zip` is created in the specified directory.

2. In IBM App Connect Enterprise Version 11.0, use the **mqsicreatebroker** command to create a new integration node.

```
mqsicreatebroker IntegrationNode
```

3. Start the integration node by running the command `mqsistart IntegrationNode`.

4. Create the directory structure `servers\IntegrationServer` under the integration node work path (`$MQSI_WORKPATH`).

For example, on Windows:

```
C:\ProgramData\IBM\MQSI\components\IntegrationNode\servers\IntegrationServer.
```

5. Stop the integration node by running the command `mqsistop IntegrationNode`.

6. Use the **mqsextractcomponents** command to migrate the IBM Integration Bus Version 10.0 or Version 9.0 integration server and resources to IBM App Connect Enterprise Version 11.0.

All deployed resources are migrated under the /run directory:

C:\ProgramData\IBM\MQSI\components\IntegrationNode\servers\IntegrationServer\run.

During migration, the IBM Integration Bus Version 10.0 or Version 9.0 configurable services are converted to policies in IBM App Connect Enterprise Version 11.0. The policy project is in the /run subfolder in the work directory. No value was specified for the policy_project_name; therefore, policies are created in a policy project called DefaultPolicies. For example, if the FTPServer configurable service name is ftpbroker, a policy file with name ftpbroker.policyxml is created under the run\DefaultPolicies folder in the work path.

7. Start the integration node by running the command `mqsistart V11BRK`.

You might see the following error message in the event viewer:

```
File node "File Output" in message flow "FileInOutFlow". The remote user identifier supplied as "myftp" is invalid.
The user identifier supplied by a securityIdentity is not valid. Either the user identifier is missing, or no securityIdentity definition exists, or the securityIdentity registry information could not be read due to a permissions problem. FTP processing for this node has been disabled.
Ensure that the securityIdentity is correctly defined using the mqsisetdbparms command.
```

Currently, the `mqsiaextractcomponents` command does not migrate the credentials that are set by using the `mqsisetdbparms` command in IBM Integration Bus Version 10.0 or Version 9.0. Therefore, you need to set the security identity and other credentials by running the `mqsisetdbparms` again in IBM App Connect Enterprise Version 11.0:

```
mqsisetdbparms V11BRK -n ftp::myftp -u user -p password
```

8. Restart the IBM App Connect Enterprise Version 11.0 integration node by running the `mqsistop` command, followed by the `mqsistart` command.

Migrating an integration node with multiple integration servers

Use the `mqsiaextractcomponents` command to migrate an IBM Integration Bus Version 10.0 or Version 9.0 integration node with multiple integration servers to IBM App Connect Enterprise Version 11.0 independent integration servers.

Procedure

1. Create an application with a few message flows in the IBM Integration Bus Version 10.0 or Version 9.0 Toolkit.
2. Create an integration node called Migration Node.
3. Create three integration servers called MigrationServer1, MigrationServer2, and MigrationServer3.
4. Deploy the application to all of the three integration servers.
5. Back up the integration node by using the `mqsibackupbroker` command:

```
mqsibackupbroker MigrationNode -d C:\temp\
```

6. Copy the broker backup file to the IBM App Connect Enterprise Version 11.0 server for migration.
7. In IBM App Connect Enterprise Version 11.0, use the `mqsiaextractcomponents` command to create three independent integration servers:

```
mqsiaextractcomponents --source-integration-node MigrationNode --source-integration-server MigrationServer1 --backup-file C:\temp\MigrationNode_181114_120112.zip --target-work-directory C:\temp\MigrationServer1 --default-application DefApp
mqsiaextractcomponents --source-integration-node MigrationNode --source-integration-server MigrationServer2 --backup-file C:\temp\MigrationNode_181114_120112.zip --target-work-directory C:\temp\MigrationServer2 --default-application DefApp
mqsiaextractcomponents --source-integration-node MigrationNode --source-integration-server MigrationServer3 --backup-file C:\temp\MigrationNode_181114_120112.zip --target-work-directory C:\temp\MigrationServer3 --default-application DefApp
```

Note: Specify the `--default-application DefApp` option to ensure that any message flows that are deployed as independent projects are moved into the default application DefApp. In IBM App Connect Enterprise Version 11.0, all resources must be part of an application.

8. Start the independent integration server by using the **IntegrationServer** command:

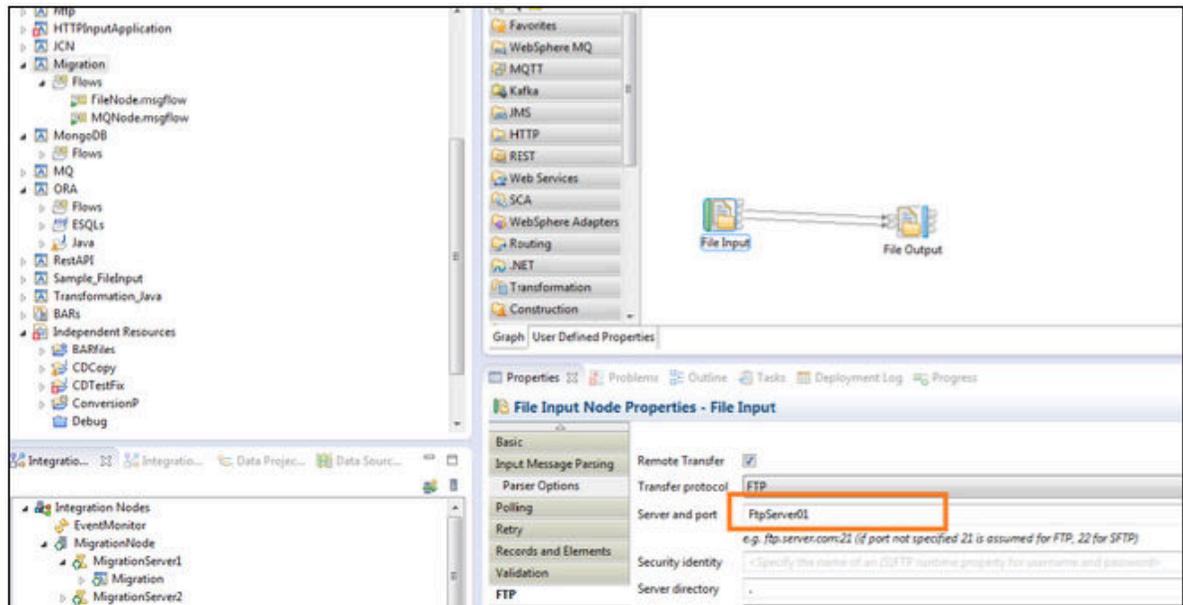
```
IntegrationServer --name MigrationServer1 --work-dir C:\temp\MigrationServer1
```

9. To demonstrate that the configurable services that were defined for the IBM Integration Bus Version 10.0 or Version 9.0 integration nodes are copied with the migrated integration server, complete the following steps.

- a) Create a configurable service.

```
mqsicreateconfigurable-service MigrationNode -c FtpServer -o FtpServer01  
-n serverName,scanDelay,transferMode,connectionType,securityIdentity -v  
one.hursley.abc.com:123,20,Binary,ACTIVE,secId
```

- b) Modify your message flow to use the FTP configurable service.



- c) Save the message flow and redeploy the application to the integration server MigrationServer1.
- d) Create a new backup by using the **mqsibackupbroker** command in IBM Integration Bus Version 10.0 or Version 9.0.

```
mqsibackupbroker MigrationNode -d C:\temp\
```

- e) In IBM App Connect Enterprise Version 11.0, run the **mqsextractcomponents** command:

```
mqsextractcomponents --source-integration-node MigrationNode --source-integration-server  
MigrationServer1 --backup-file C:\temp\MigrationNode_181114_125348.zip --target-work-  
directory C:\temp\MigrationServer4 --default-application DefApp  
BIP8071I: Successful command completion.
```

The folder `C:\temp\MigrationServer4\run\DefaultPolicies` contains the FTP configurable service policy under the IBM App Connect Enterprise Version 11.0 independent integration server. The `FtpServer01` configurable service is available as a new policy. The `FtpServer01.policyxml` file contains configurations that match your FTP Server configurations.

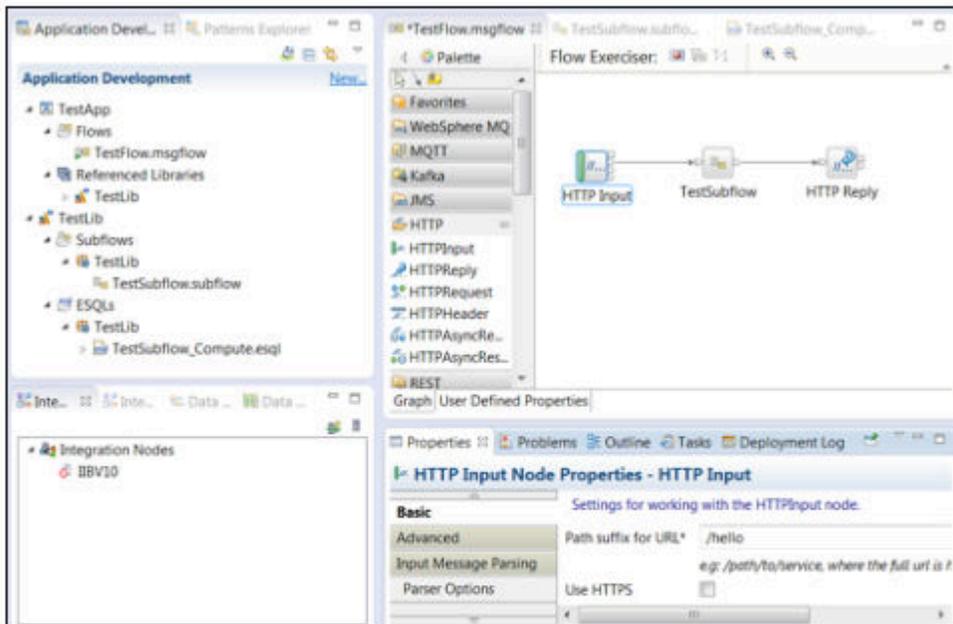
- f) Start the independent integration server by using the **IntegrationServer** command and verify the flow execution.

Migrating an integration server that contains applications and a shared library

Migrate an IBM Integration Bus Version 10.0 or Version 9.0 integration server that contains applications and a shared library to IBM App Connect Enterprise Version 11.0.

Procedure

1. In IBM Integration Bus Version 10.0 or Version 9.0, create a shared library to contain a simple subflow. Create an application with a simple message flow that uses the subflow from the shared library.



```
CREATE COMPUTE MODULE TestSubflow_Compute
CREATE FUNCTION Main() RETURNS BOOLEAN
BEGIN
    CALL CopyMessageHeaders();
    -- CALL CopyEntireMessage();
    SET OutputRoot.XMLNSC.out = 'Hello ' || InputRoot.XMLNSC.in;
    RETURN TRUE;
END;
```

2. Deploy the shared library to the IBM Integration Bus Version 10.0 or Version 9.0 integration node.
3. Back up the IBM Integration Bus Version 10.0 or Version 9.0 integration node by using the **mqsibbackupbroker** command.

```
mqsibbackupbroker IIBV10 -d c:\temp
```

4. Create a new IBM App Connect Enterprise Version 11.0 integration node called by using the command **mqsicreatebroker ACEV11**.
5. Start the IBM App Connect Enterprise Version 11.0 node to set up the work directory.
6. Stop the IBM App Connect Enterprise Version 11.0 node.
7. Create a blank work directory for the integration server IS1 to be migrated to IBM App Connect Enterprise Version 11.0.
8. Use the **mqsextractcomponents** command to migrate the IBM Integration Bus Version 10.0 or Version 9.0 integration server IS1 to IBM App Connect Enterprise Version 11.0, then start the Version 11.0 integration node.

```
mqsextractcomponents --source-integration-node IIBV10 --source-integration-server IS1 --
backup-file c:\temp\IIBV10_181118_220025.zip
--target-work-directory c:\ProgramData\IBM\MQSI\components\ACEV11\servers\IS1
```

```
mqsistart ACEV11
```

9. After you start the node, connect to the integration node from the IBM App Connect Enterprise Toolkit.

The application and shared library resources that were migrated from the IBM Integration Bus Version 10.0 or Version 9.0 node appear under the Version 11.0 integration node.

10. Test the migrated flow.
For example:

```
C:\user\curl_7_53_1_openssl_nghttp2_x64>curl -d "<in>Test</in>" http://localhost:7800/hello
<out>Hello Test</out>
```

Migrating an SAP integration flow

Migrate an SAP integration flow that runs on IBM Integration Bus Version 10.0 or Version 9.0 to IBM App Connect Enterprise Version 11.0.

About this task

The following steps describe how to create an SAP integration flow in IBM Integration Bus Version 10.0 or Version 9.0, then how to migrate it to IBM App Connect Enterprise Version 11.0. The steps describe two different migration options: migrating to an integration node or an independent integration server. This scenario uses an SAP outbound adapter as an example; the same steps apply to an inbound adapter.

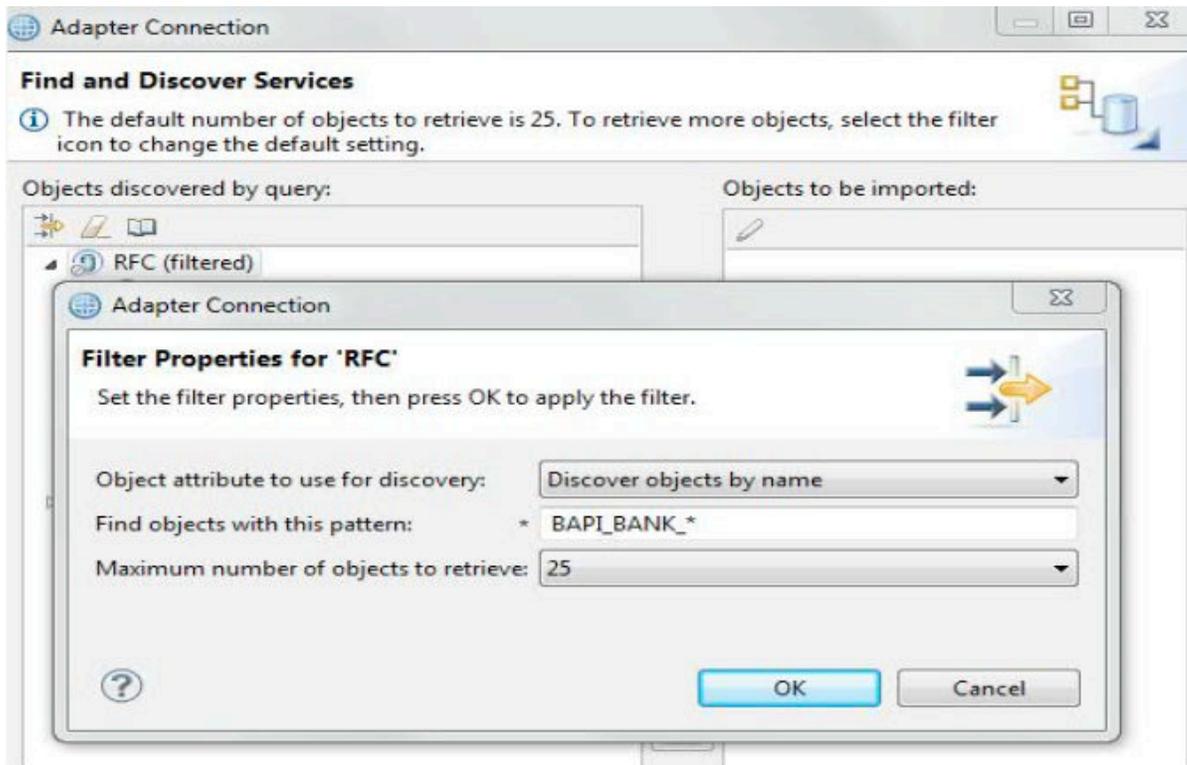
Procedure

- [“Creating an SAP integration flow in IBM Integration Bus Version 10.0 or Version 9.0” on page 16](#)
- [“Migrating an IBM Integration Bus Version 10.0 or Version 9.0 integration node to an IBM App Connect Enterprise Version 11.0 integration node” on page 21](#)
- [“Migrating an integration flow from an IBM Integration Bus Version 10.0 or Version 9.0 integration node to an IBM App Connect Enterprise Version 11.0 independent integration server” on page 22](#)

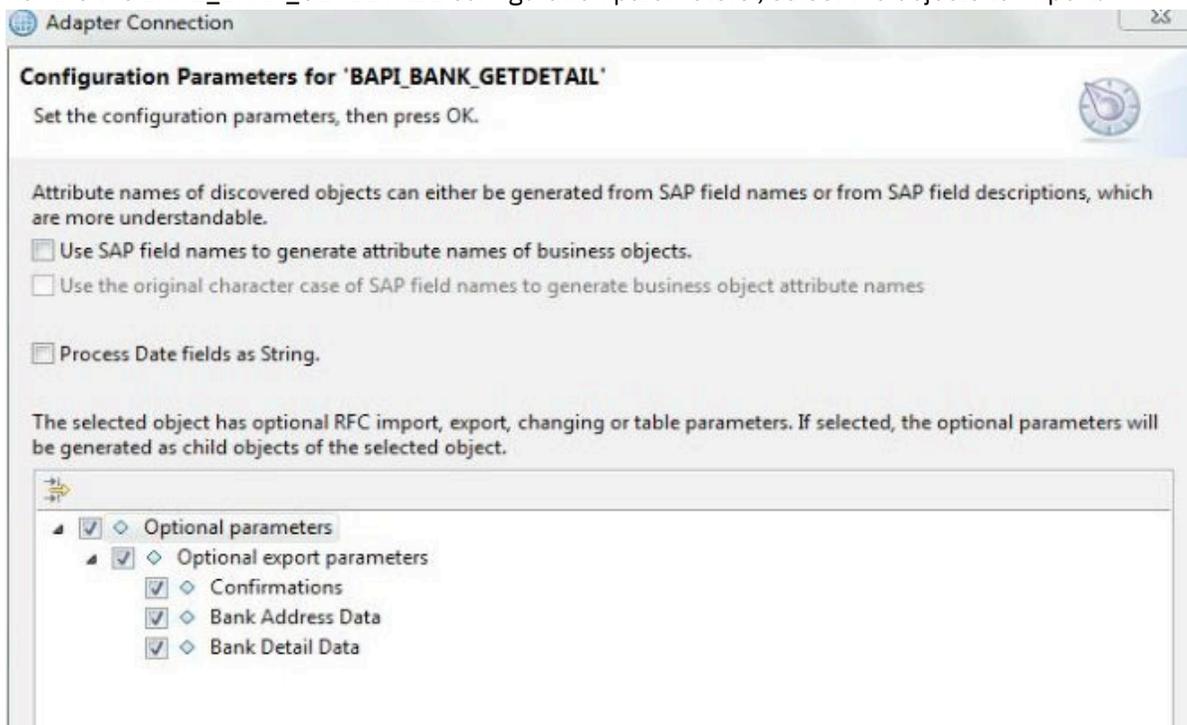
Creating an SAP integration flow in IBM Integration Bus Version 10.0 or Version 9.0

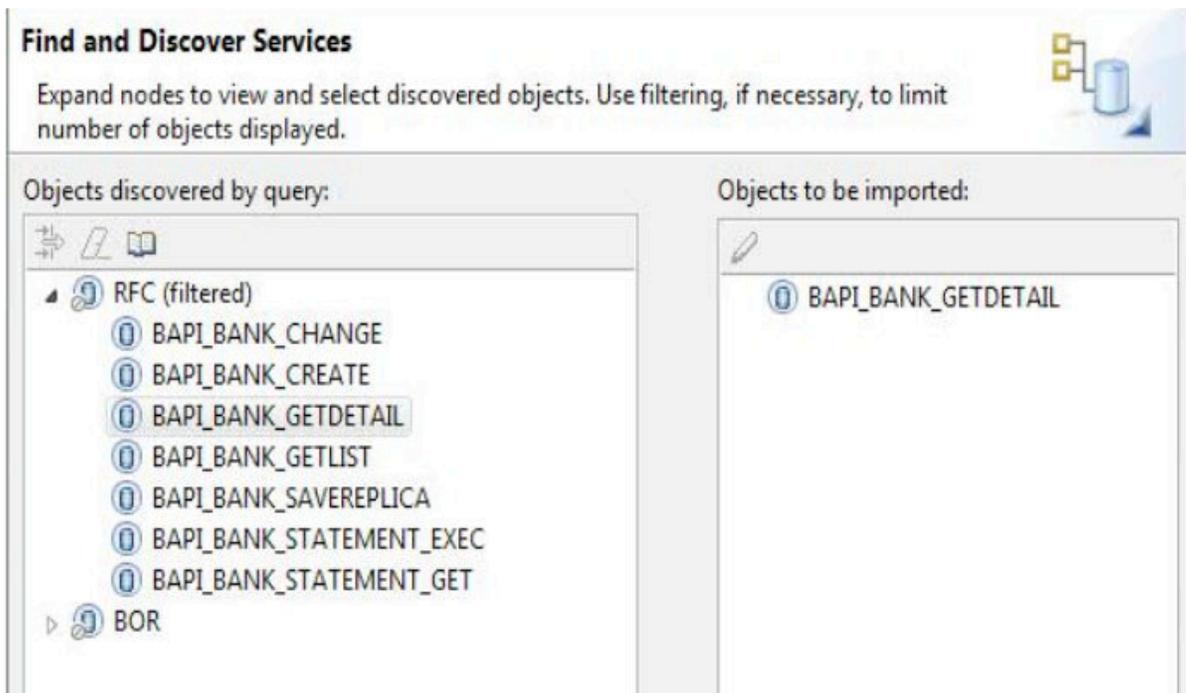
Procedure

1. In IBM Integration Bus Version 10.0 or Version 9.0, create an SAP outbound adapter to fetch BAPI_BANK_GETDETAILS.
 - a) In the IBM Integration Toolkit, click **File > New > Adapter connection**, then follow the instructions in the wizard.
 - b) For service discovery, select **RFC** and set a filter to find objects for BAPI_BANK_*

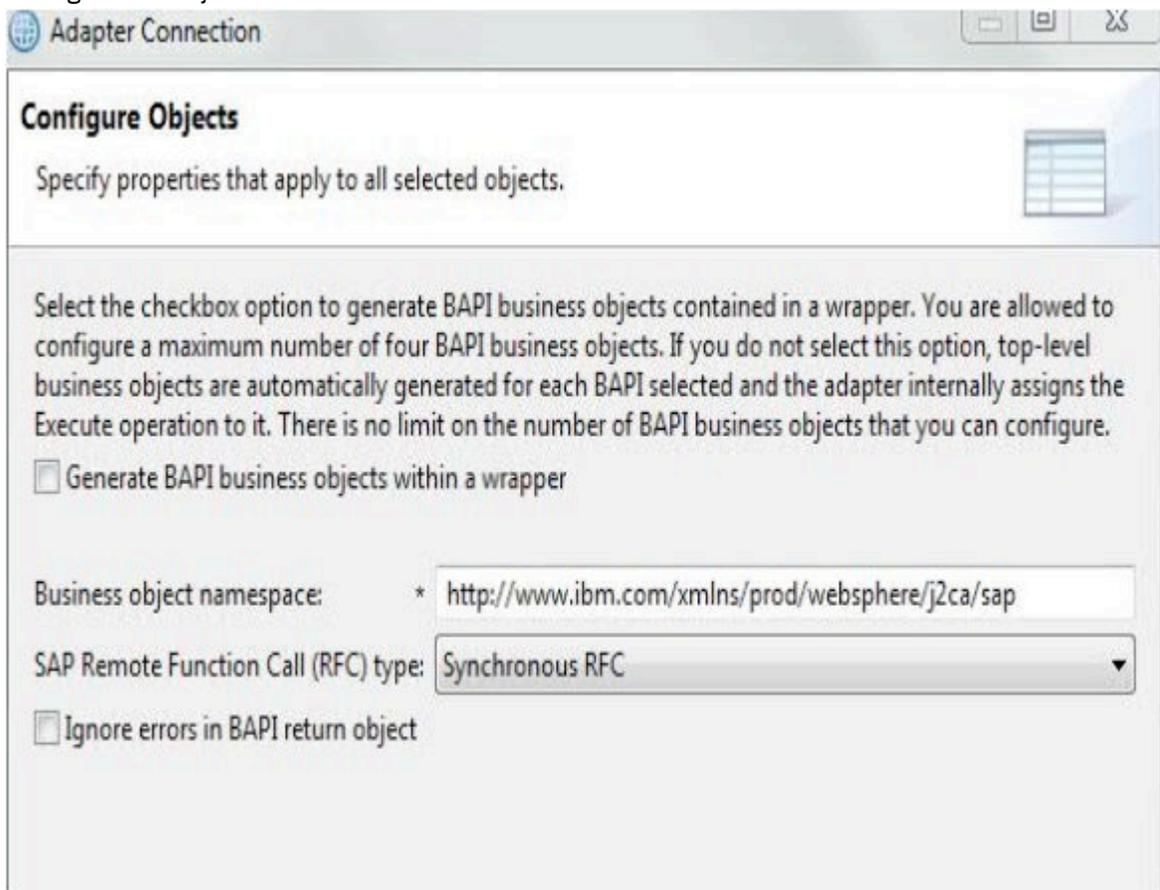


c) For the the BAPI_BANK_GETDETAIL configuration parameters , select the objects to import.





d) Configure the objects.



2. Create a simple SAP application to fetch BAPI BANK GETDETAILS.
3. Configure the integration node with SAP JCo libraries by using the **mqsichangeproperties** command.

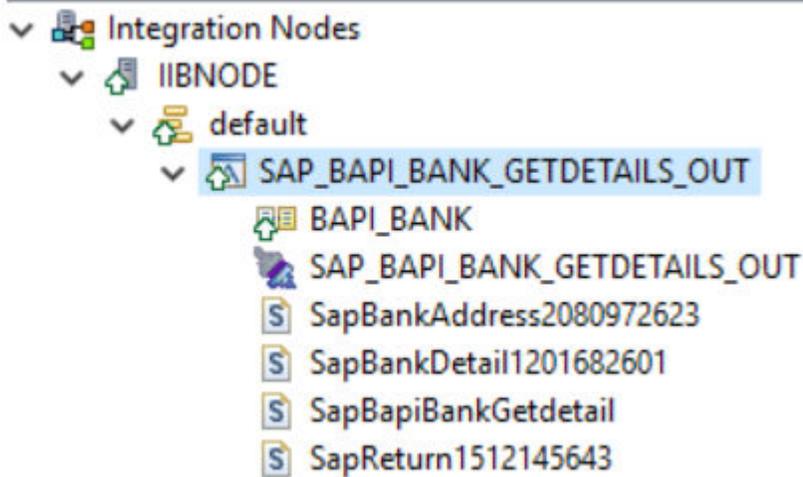
```
mqsichangeproperties IIBNODE -c EISProviders -o SAP -n jarsURL -v C:\SAP_JARS
mqsichangeproperties IIBNODE -c EISProviders -o SAP -n nativeLibs -v C:\SAP_JARS
```

You can verify that the properties are set up correctly by running the command:

```
mqsireportproperties IIBNODE -c EISProviders -o SAP -r
```

```
C:\Program Files\IBM\IIB\10.0.0.16>mqsireportproperties IIBNODE -c EISProviders -o SAP -r
EISProviders
SAP
  jarsURL='C:\SAP_JARS'
  nativeLibs='C:\SAP_JARS'
BIP8071I: Successful command completion.
```

4. Deploy the resources to the integration server.



5. Create a configurable service for the SAP adapters connection.

By using a configurable service, you can change the host name that is set in the adapter file without needing to redeploy the resources.

a) Use the SAPConnection configurable service to change connection details for an SAP adapter.

```
mqsicreateconfigurable IIBNODE -c SAPConnection -o
SAP_BAPI_BANK_GETDETAILS_OUT.outadapter -n applicationServerHost,client -v
wmsap.hursley.ibm.com,001
```

b) To display all SAPConnection configurable services, use the **mqsireportproperties** command:

```
mqsireportproperties IIBNODE -c SAPConnection -o AllReportableEntityNames -r
```

```
C:\Program Files\IBM\IIB\10.0.0.16>mqsireportproperties IIBNODE -c SAPConnection -o AllReportableEntityNames -r
SAPConnection
Template
  RFCTraceLevel=''
  RFCTraceOn=''
  RFCTracePath=''
  SAPSystemID=''
  applicationServerHost=''
  assuredOnceDelivery=''
  client=''
  gatewayHost=''
  gatewayService=''
  loadBalancing=''
  logonGroup=''
  messageServerHost=''
  numberOfListeners=''
  retryConnectionOnStartup=''
  retryInterval=''
  retryLimit=''
  rfcProgramID=''
  sharedTidStoreClientDefinitionFile=''
  sharedTidStoreMgr=''
  systemNumber=''
  SAP_BAPI_BANK_GETDETAILS_OUT.outadapter
  RFCTraceLevel=
  RFCTraceOn=''
  RFCTracePath=''
  SAPSystemID=''
  applicationServerHost='wmsap.hursley.ibm.com'
  assuredOnceDelivery=''
  client='001'
  connectionIdleTimeout='0'
  gatewayHost=''
```

Migrating an IBM Integration Bus Version 10.0 or Version 9.0 integration node to an IBM App Connect Enterprise Version 11.0 integration node

Procedure

1. Back up the IBM Integration Bus Version 10.0 or Version 9.0 integration node by using the **mqsibbackupbroker** command.

```
mqsibbackupbroker IIBNODE -d C:\SAP_JARS
```

2. Migrate the IBM Integration Bus Version 10.0 or Version 9.0 integration node and resources to IBM App Connect Enterprise Version 11.0 by using the **mqsextractcomponents** command.

```
mqsextractcomponents --source-integration-node IIBNODE --target-integration-node ACENODE --backup-file "C:\SAP_JARS\IIBNODE_190712_083247"
```

This **mqsextractcomponents** command creates a new integration node and integration server. If the integration node already exists, the **mqsextractcomponents** command fails unless you specify `-delete-existing-node`. In this case, the existing integration node is deleted and a new integration node is created with the same name.

All deployed resources are migrated under the `/run` directory of the created integration server. For example, on Windows:
`C:\ProgramData\IBM\MQSI\components\IntegrationNode\servers\IntegrationServer.`

During migration, the IBM Integration Bus Version 10.0 or Version 9.0 configurable services are converted to policies in IBM App Connect Enterprise Version 11.0. You can find the policy project in the integration node work directory. If you do not specify a policy project name, policies are created in a policy project called `DefaultPolicies`. The policy file that is created has the same name as the adapter.

3. Start the IBM App Connect Enterprise Version 11.0 integration node by running the command `mqsistart ACENODE`.

The application and policy files are listed under the IBM App Connect Enterprise Version 11.0 node in the Toolkit. You can confirm the location of the migrated JAR files by running the **mqsireportproperties** command on the IBM App Connect Enterprise Version 11.0 node.

```
mqsireportproperties ACENODE -c EISProviders -o SAP -r
```

4. Test the integration flow by sending a message to fetch the record for BAPI BANK GETDETAILS. The following example shows the expected output message.

```
<NS1:SapBapiBankGetdetail xmlns:NS1="http://www.ibm.com/xmlns/prod/websphere/12ca/sap/sapbapibankgetdetail">
  <SapBankAddress>
    <NameOfBank>Laan og Spar</NameOfBank>
    <HouseNumberAndStreet>Havnegade 6</HouseNumberAndStreet>
    <City>1900 Copenhagen V</City>
    <BankNumber>1200</BankNumber>
  </SapBankAddress>
  <SapBankDetail>
    <DateOnWhichTheRecordWasCreated>1997-10-02</DateOnWhichTheRecordWasCreated>
    <NameOfPersonWhoCreatedTheObject>SAP</NameOfPersonWhoCreatedTheObject>
  </SapBankDetail>
  <SapReturn>
    <MessageNumber>000</MessageNumber>
    <ApplicationLogInternalMessageSerialNumber>000000</ApplicationLogInternalMessageSerialNumber>
    <LinesInParameter>0</LinesInParameter>
  </SapReturn>
</NS1:SapBapiBankGetdetail>
```

Migrating an integration flow from an IBM Integration Bus Version 10.0 or Version 9.0 integration node to an IBM App Connect Enterprise Version 11.0 independent integration server

Procedure

1. Migrate an IBM Integration Bus Version 10.0 or Version 9.0 integration server to an IBM App Connect Enterprise Version 11.0 independent integration server by running the **mqsextractcomponents** command.

```
mqsextractcomponents --source-integration-node IIBNODE --source-integration-server default --target-work-directory "C:\temp\ACESIS" --backup-file "C:\SAP_JARS\IIBNODE_190712_083247.zip"
```

If the work directory already exists, the **mqsextractcomponents** command fails unless you specify `-clear-work-directory`. In this case, the configuration and resources are written to the work directory, and overwrite any data that might be present in the directory.

2. Start the IBM App Connect Enterprise Version 11.0 integration server by running the **IntegrationServer** command.

```
IntegrationServer --name default --work-dir C:\temp\ACESIS
```

All deployed resources are migrated to the `/run` subfolder in the work directory of the independent integration server.

During migration, the values that are set by using the **mqsichangeproperties** command in IBM Integration Bus Version 10.0 or Version 9.0 are added to the `server.conf.yaml` file in the `/overrides` subfolder in the work directory of the IBM App Connect Enterprise Version 11.0 integration server.

Also during migration, the IBM Integration Bus Version 10.0 or Version 9.0 configurable services are converted to policies in IBM App Connect Enterprise Version 11.0. You can find the policy project in the `/run` subfolder of the work directory for the integration server. If you do not specify the policy project name, policies are created in a policy project called `DefaultPolicies`. The name of the policy file that is created is the same as the adapter.

The application and policy files are listed under the IBM App Connect Enterprise Version 11.0 integration server in the Toolkit.

Performing pre-migration tasks

Perform the prerequisite tasks for migration to IBM App Connect Enterprise Version 11.0.

About this task

Complete the following steps before you migrate your integration nodes or integration servers:

- Back up your resources; see [“Backing up resources before migration” on page 23](#).
- If your environment includes access to databases, create ODBC definitions for the databases and specify appropriate database drivers; see [“Updating ODBC definitions when you migrate to IBM App Connect Enterprise Version 11.0” on page 23](#).
- (Optional) If you plan to move your architecture to adopt containers, run the Transformation Advisor tool; see [“Running the Transformation Advisor tool” on page 25](#).

What to do next

After you complete your pre-migration tasks, perform the migration tasks by following the instructions in [“Migrating to IBM App Connect Enterprise Version 11.0” on page 40](#).

Backing up resources before migration

Back up your resources before you start to migrate to IBM App Connect Enterprise Version 11.0.

Before you begin

To plan your migration strategy, read [“Preparing for migration” on page 1](#).

About this task

Before you complete migration tasks, back up your current resources by completing the following steps.

Procedure

1. Optional: If your message flows access user databases through an ODBC connection, back up the ODBC files that you use for these connections.
Take a copy of these files and store them safely in a different location.
2. Back up your IBM Integration Toolkit workspace and resources; for example, message flow files, message set definition files, Java files, ESQL files, mapping files, XML Schema files, and BAR files.
 - Export all your projects from your current IBM Integration Toolkit.
 - Archive your workspace resources:
 - If you manage your workspace resources in a shared repository, for example CVS, follow standard backup procedures for safeguarding versions. Create a version for storing Version 11.0 resources.
 - If you maintain your workspace resources on a local or shared disk, copy your workspace directory to a different location.

What to do next

After you have backed up your resources, update your ODBC definitions by following the instructions in [“Updating ODBC definitions when you migrate to IBM App Connect Enterprise Version 11.0” on page 23](#).

Updating ODBC definitions when you migrate to IBM App Connect Enterprise Version 11.0

Before you migrate an integration server or integration node, create ODBC definitions for databases and specify appropriate database drivers for IBM App Connect Enterprise Version 11.0.

About this task

The database drivers that are supported by IBM App Connect Enterprise Version 11.0 might be at a later version than the drivers used by previous versions.

Follow the instructions that are provided for your operating system:

Windows systems

Change the ODBC connection definitions:

1. Open the **ODBC Data Source Administrator** window.
2. Open the **System DSN** page.
3. For each Oracle and Sybase database that is accessed by the integration node, compare the ODBC driver against the entries that are listed in the following table. If the ODBC driver does not match, you need to associate the data source name with the new ODBC driver by using the following instructions:
 - a. Delete the data source by clicking **Remove**.
 - b. Re-create the data source with the new ODBC driver by clicking **Add**.

The following table displays the name of the new ODBC driver for each database management system (DBMS).

DBMS	IBM App Connect Enterprise Version 11.0 ODBC driver name
Sybase	IBM App Connect Enterprise 11.0.0. <i>n</i> - DataDirect Technologies 64-BIT Sybase Wire Protocol
Oracle	IBM App Connect Enterprise 11.0.0. <i>n</i> - DataDirect Technologies 64-BIT Oracle Wire Protocol

where *n* is the level of the installed fix pack.

Change the XA resource manager definitions:

1. Open the **Properties** window of the integration node queue manager by using the IBM MQ Services snap-in.
2. Open the **Resources** page.
3. For each Oracle and Sybase database that participates in a global unit of work that is coordinated by the integration node queue manager, change the contents of the **SwitchFile** field. For changes to the switch file configuration to take effect, you must restart the integration node queue manager.

The following table specifies the entries you must change for each database management system (DBMS). *install_dir* represents the fully qualified path name of the directory in which you installed IBM App Connect Enterprise.

DBMS	Original entry	New entry
Oracle	<i>install_dir</i> \server\bin\ukor8dttc22.dll or <i>install_dir</i> \server\bin\ukor8dttc23.dll or <i>install_dir</i> \server\bin\ukora24.dll or <i>install_dir</i> \server\bin\ukora26.dll	WBIMB\bin\ukora95.dll
Sybase	<i>install_dir</i> \server\bin\ukase22.dll or <i>install_dir</i> \server\bin\ukase23.dll or <i>install_dir</i> \server\bin\ukase24.dll or <i>install_dir</i> \server\bin\ukase26.dll	WBIMB\bin\ukase95.dll

Linux systems

Change the ODBC connection definitions:

Create an ODBC definitions file by following the instructions in [Connecting to a database from and systems by using the](#). Before you run the commands at the new service level, check that your ODBCINI environment variable points to the new file and not to the existing file. Check that ODBCYSINI environment variable is set to point to the directory that contains your odbcinst.ini file.

Change the XA resource manager definitions:

To change the XA resource manager definitions, edit the queue manager configuration file (`qm.ini`) of the queue manager that is associated with the integration node. The `qm.ini` file is in the directory `/var/mqm/qmgrs/queue_manager_name`, where `queue_manager_name` is the name of the queue manager that is associated with the integration node.

In the `XAResourceManager` stanza for each Oracle and Sybase database that participates in a global unit of work that is coordinated by the integration node queue manager, change the entry for the switch file. For changes to the switch file configuration to take effect, you must restart the integration node queue manager.

The following table specifies the entries that you must change for each operating system and database management system (DBMS).

DBMS	Original entry	New entry
Oracle	SwitchFile=UKor8dttc22.so or SwitchFile=UKoradttc22.so or SwitchFile=UKor8dttc23.so or SwitchFile=UKoradttc23.so or SwitchFile=UKoradttc24.so or SwitchFile=UKoradttc26.so	SwitchFile=UKoradttc95.so
Sybase (not supported on Linux on Z)	SwitchFile=UKasedttc22.so or SwitchFile=UKasedttc23.so or SwitchFile=UKasedttc24.so or SwitchFile=UKasedttc26.so	SwitchFile=UKasedttc95.so

Running the Transformation Advisor tool

Before you complete your migration tasks, collect data about what was deployed to your IBM Integration Bus Version 10.0 integration node, or your IBM App Connect Enterprise Version 11.0 integration node. If you plan to move your architecture to adopt containers, analyze the data that you collected for potential issues.

Before you begin

To plan your migration strategy, read [“Preparing for migration” on page 1](#).

About this task

Analyze the data that you collected about your IBM Integration Bus Version 10.0 integration node, or your IBM App Connect Enterprise Version 11.0 integration node by completing the following steps.

Procedure

1. Back up the resource that you want to migrate by completing one of the following steps:
 - Create a backup file of your IBM Integration Bus Version 10.0 integration node, or your IBM App Connect Enterprise Version 11.0 integration node by running the **mqsibbackupbroker** command:

```
mqsibbackupbroker node10 -d C:\temp -a node10.zip
```

- Add the resources that you want to migrate to a deployable BAR file by running the **mqsipackagebar** command:

```
mqsipackagebar -w C:\Workspace -a myapp.bar -c -i -k Application1 -y Shlib1 -v tracefile
```

2. Optional: Create an output directory where the logs from the **TADDataCollector** command are written, and set the value of the environment variable *TADDataCollectorDirectory* to refer to it. If you choose not to complete this step, the logs are written to a temporary folder in the home directory of the user who runs the command.

- a) Create an output directory where the logs from the **TADDataCollector** command are written.

```
mkdir C:\TADemo
```

- b) Set the value of the environment variable *TADDataCollectorDirectory* to the path of the output directory you created at step “2.a” on page 26.

```
set TADDataCollectorDirectory=C:\TADemo
```

3. Required: Run the **TADDataCollector** command with one of the following options:

- a) Optional: To collect data:

```
TADDataCollector ace collect C:\temp\node10.zip
```

- b) Optional: To collect and assess data:

```
TADDataCollector ace assess C:\temp\node10.zip
```

- c) Optional: To generate reports on data that is collected at step “3.a” on page 26 and assessed at step “3.b” on page 26:

```
TADDataCollector ace report
```

- d) Optional: To collect and assess data, and generate reports:

```
TADDataCollector ace run C:\temp\node10.zip
```

For more information about the **TADDataCollector** command, see [command](#).

4. Go to the output directory where the output from the **TADDataCollector** command is written.

If you completed step “2.a” on page 26 and step “2.b” on page 26, the output is written to the directory referenced by the environment variable *TADDataCollectorDirectory*, in this example, C:\TADemo. Otherwise, the output is written to a temporary folder in the home directory of the user who runs the command. For example, C:\Users\KEVINM~1\AppData\Local\Temp\TADDataCollector.

The following subdirectories are created in the output directory:

- logs This folder contains the log *ta_util.log* that you can use to check for errors in the process.
- output This folder contains
 - The file *environment.json*, showing the details of the environment where the integration node was created.
 - A sub folder for each integration server, for example *node10/server10*, containing a *.json* file. The *.json* file contains details of what is deployed to the integration server. It also

contains configuration details that were generated when you ran the command at step “3” on page 26.

- Go to the output folder for the IBM Integration Bus Version 10.0 that you want to review. For example, C:\TADemo\output\node10.

If you run the **TADDataCollector** command with the `run` parameter, a static HTML report, C:\TADemo\output\node10\recommendations.html is produced. The report lists any issues that are found for each integration server under the integration node.

Follow the links in the summary table, or scroll down the page, to view more detailed information about the issues that are found in each integration server. The *Overall Complexity Score* is assessed as either Simple, Moderate, or Complex and the following guidelines apply:

<i>Table 1. Overall Complexity</i>	
Complexity	Action
Simple	Admin change is required.
Moderate	Development change is required.
Complex	Difficult development task or an alternative strategy is required.

The Transformation Advisor tool also provides a severity classification for each issue it uncovers:

<i>Table 2. Severity Classification</i>	
Severity Classification	Action
GREEN (INFO)	No immediate action is required, but you might want to be aware.
YELLOW (WARNING)	Immediate action is probably required or advised before you proceed.
RED (ERROR)	You cannot proceed without taking remedial action.

The following example shows a Complex issue with a severity classification of RED that was identified when the tool was run against the IBM Integration Bus Version 10.0 integration node, *node10* with one integration server, *server10*.

<i>Table 3. Recommendation report for assessment: node10</i>			
Assessment unit	Overall complexity score	Issues	Total effort (days)
server10	COMPLEX	1	10

<i>Table 4. Recommendation report for assessment unit: server10</i>				
Product name	Product version	Runtime	Platform	Location
ACE	11.0	ACE	Docker	Private

<i>Table 5. Overall complexity score: COMPLEX</i>		
RED issues: 1	Yellow issues: 0	Green issues: 0

Table 6. Issues with a COMPLEX complexity rating can be resolved by making significant development changes or by choosing an alternate technology:

ID	Title	Cost	Severity	Solution
IIB01	Consider a different transformation mechanism in place of .NET	10	RED	<p>The message flow has been found to contain an instance of a .NETInput or .NETCompute message flow node.</p> <p>Whilst App Connect Enterprise v11 software continues to support .NET, there is no support for running the .NET CLR when deploying to Linux Docker containers on App Connect Enterprise Certified Containers.</p> <p>Other message flow nodes are available for transformation such as Compute nodes, JavaCompute nodes, and Mapping nodes.</p>

For more information about the rules that apply to the Transformation Advisor tool, see [“Rules for the Transformation Advisor tool”](#) on page 29.

6. Complete any actions that are advised by the report in step [“5”](#) on page 27 and run the Transformation Advisor tool again to confirm that all issues are resolved.

What to do next

Complete any remaining pre-migration tasks; see [“Performing pre-migration tasks”](#) on page 22.

Complete the migration tasks by following the instructions in [“Migrating to IBM App Connect Enterprise Version 11.0”](#) on page 40.

Rules for the Transformation Advisor tool

The following table shows the rules that apply to the Transformation Advisor tool and whether they apply to a backup file, a bar file, or both.

<i>Table 7. Rules for the Transformation Advisor tool:</i>				
ID	Title	Can be found in backup file	Can be found in BAR file	Solution
II B 0 1	Consider a different transformation mechanism in place of .NET.	True	True	<p>The message flow contains an instance of a .NETInput or a .NETCompute message flow node.</p> <p>While IBM App Connect Enterprise v11 software continues to support .NET, there is no support for running the .NET CLR when it is deployed to Linux. Docker containers on App Connect Enterprise Certified Containers.</p> <p>Other message flow nodes are available for transformation such as Compute nodes, JavaCompute nodes, and Mapping nodes.</p>
II B 0 2	Consider a different transformation mechanism in place of PHP.	True	True	<p>The message flow contains an instance of a PHPCompute message flow node. The PHPCompute node was deprecated in IBM Integration Bus v10, and it was removed from IBM App Connect Enterprise v11. Other message flow nodes are available for transformation, such as Compute nodes, JavaCompute nodes, and Mapping nodes.</p>
II B 0 3	Consider an alternative mechanism to SCA to communicate with WebSphere Process Server.	True	True	<p>The message flow contains an instance of an SCA message flow node (SCAInput, SCAREply, SCAREquest, SCAAsyncRequest, SCAAsyncResponse). IBM App Connect Enterprise v11 does not support the SCA message flow nodes that were available in IBM Integration Bus v10. HTTP, IBM MQ, or JMS are all still viable transport options for communication between message flows and SCA components in WebSphere Process Server.</p>

Table 7. Rules for the Transformation Advisor tool: (continued)

ID	Title	Can be found in backup file	Can be found in BAR file	Solution
II B 0 4	Consider a different mechanism to run IBM Operational Decision Management Business Rules.	True	True	The message flow contains an instance of a DecisionService message flow node. IBM App Connect Enterprise v11.0.0.8 provides a new replacement message flow node that is called the ODMRules node. The purpose of this message flow node is to run ODM rules within the integration server. Alternatively, you can use the ODM SOAP or REST API to start the execution of business rules in the ODM engine.
II B 0 5	Consider a different protocol rather than relying on local file integration.	True	True	The message flow contains an instance of a FileInput message flow node that relies on local file interaction and is not configured to use FTP. Although IBM App Connect Enterprise v11 continues to support reading local files with the FileInput node, for users who embrace a container-based architecture, this choice has architectural drawbacks. Consider changing your configuration to use FTP, or a more suitable messaging-based transport.
II B 0 6	Consider your use of WebSphere Service Registry and Repository.	True	True	The message flow contains an instance of a RegistryLookup or EndpointLookup message flow node that communicates with WebSphere Service Registry and Repository. Releases before IBM App Connect Enterprise v11.0.0.7 do not support these message flow nodes. If you have an urgent need, or a simple use case, you can use the SOAP nodes for communication with WebSphere Service Registry and Repository. Alternatively, contact IBM to learn more about future plans in this area.

Table 7. Rules for the Transformation Advisor tool: (continued)

ID	Title	Can be found in backup file	Can be found in BAR file	Solution
II B 0 7	A Publication message flow node was found. You might want to consider altering your IBM MQ topology.	True	True	The message flow contains an instance of a Publication message flow node. In IBM Integration Bus v10, and in IBM App Connect Enterprise v11.0 before Fix Pack 6, this message flow node requires a local server binding connection to an IBM MQ Queue Manager. IBM App Connect Enterprise v11.0.0.6 supports the Publication node by using a remote client connection to an IBM MQ Queue Manager. Consider altering your IBM MQ topology as part of your move to a container-based architecture.
II B 0 8	A Sequence or Resequence message flow node was found. Consider altering your IBM MQ topology.	True	True	The message flow contains an instance of a Sequence or Resequence message flow node. In IBM App Connect Enterprise before v11.0.0.7, this message flow node requires a local server binding connection to an IBM MQ Queue Manager. Consider altering your IBM MQ topology as part of your move to a container-based architecture.
II B 0 9	A Collector message flow node was found. Consider altering your IBM MQ topology.	True	True	The message flow contains an instance of a Collector message flow node. In IBM App Connect Enterprise before v11.0.0.7, this message flow node requires a local server binding connection to an IBM MQ Queue Manager. Consider altering your IBM MQ topology as part of your move to a container-based architecture.
II B 1 0	A TimeoutControl or TimeoutNotification message flow node was found. You might want to consider altering your IBM MQ topology.	True	True	The message flow contains an instance of a TimeoutControl or TimeoutNotification message flow node. In IBM App Connect Enterprise before v11.0.0.7, these message flow nodes require a local server binding connection to an IBM MQ Queue Manager. Consider altering your IBM MQ topology as part of your move to a container-based architecture.

Table 7. Rules for the Transformation Advisor tool: (continued)

ID	Title	Can be found in backup file	Can be found in BAR file	Solution
II B 1 1	An AggregateControl, AggregateRequest, or AggregateReply message flow node was found. You might want to consider altering your IBM MQ topology.	True	True	The message flow contains an instance of an AggregateControl, AggregateRequest, or AggregateReply message flow node. In IBM App Connect Enterprise before v11.0.0.7, these message flow nodes require a local server binding connection to an IBM MQ Queue Manager. Consider altering your IBM MQ topology as part of your move to a container-based architecture. IBM App Connect Enterprise v11 also provides the Group nodes, which cater for similar aggregation use-cases but use in-memory queuing and have no IBM MQ dependency.
II B 1 2	A KafkaConsumer or KafkaProducer message flow node was found. You might want to consider changing the version of your Kafka broker.	True	True	The message flow contains an instance of a KafkaConsumer or KafkaProducer message flow node. In IBM App Connect Enterprise v11.0.0.4 (and earlier fix packs), the product uses a Kafka client at version 0.10.0.1. In IBM App Connect Enterprise v11.0.0.5 (and later fix packs), the product uses a Kafka client at version 2.20. You might want to change to a different client version when you consider the compatibility of your Kafka broker.
II B 1 3	A JDEdwardsInput, JDEdwardsRequest, PeopleSoftInput, PeopleSoftRequest, SiebelInput, or SiebelRequest message flow node was found. You might want to consider the version of your App Connect Enterprise installation.	True	True	The message flow contains an instance of a JDEdwardsInput, JDEdwardsRequest, PeopleSoftInput, PeopleSoftRequest, SiebelInput, or SiebelRequest message flow node. These message flow nodes are also supported in IBM App Connect Enterprise v11. If you used configurable service definitions with these message flow nodes, you might want to consider the introduction of JDEdwards, PeopleSoft, and Siebel policy types in IBM App Connect Enterprise v11.0.0.6.

Table 7. Rules for the Transformation Advisor tool: (continued)

ID	Title	Can be found in backup file	Can be found in BAR file	Solution
II B 1 4	An SAPInput or SAPRequest message flow node was found. You might want to consider changing the configuration of your IBM App Connect Enterprise installation.	True	True	The message flow contains an instance of an SAPInput or SAPRequest message flow node. These message flow nodes are also supported in IBM App Connect Enterprise v11, where the concept of an IBM Integration Bus configurable service is replaced with an IBM App Connect Enterprise policy. When you move to a container-based architecture, consider how to make the SAP JCo libraries available to your containers and the required settings in server.conf.yaml.
II B 1 5	A deprecated graphical mapping message flow node was found. Convert this node to the newer Mapping node.	True	True	The message flow contains an instance of the old graphical mapping message flow node. This type of message flow node is no longer supported. From WebSphere Message Broker v8.0 onwards, you must convert message maps to graphical data maps. The Toolkit provides a conversion tool for this purpose.
II B 1 6	A TCPIPServer message flow node was found. Consider altering the configuration of your containers to open the required TCPIP port.	True	True	The message flow contains an instance of a TCPIPServer message flow node. Consider changing your values.yaml file if you intend to use a non-default TCPIP port in your containers.
II B 1 7	A LoopbackRequest message flow node was found. Consider altering the configuration of your containers to support this node.	True	True	The message flow contains an instance of a LoopbackRequest message flow node. Your IBM App Connect Enterprise v11 installation provides Node Package Manager (npm) as a convenient means of configuring the required Loopback Java™ script modules to support this node. When you move to a container-based architecture, consider your build pipeline and its abilities to configure these supporting files in your container.

Table 7. Rules for the Transformation Advisor tool: (continued)

ID	Title	Can be found in backup file	Can be found in BAR file	Solution
II B 1 8	A WebSphere Transformation Extender or IBM Transformation Extender message flow node was found. Consider changing the configuration of your containers to support this node.	True	True	The message flow contains an instance of a WebSphere Transformation Extender or IBM Transformation Extender message flow node. IBM App Connect Enterprise v11.0.0.4 (and later) supports the use of this type of message flow node, which is used with IBM Transformation Extender v10. When you move to a container-based architecture, consider this version information.
II B 1 9	An MQInput, MQOutput, or MQGet message flow node that uses server bindings to a queue manager was found. You might want to consider changing the node to use IBM MQ Client bindings when you move to containers.	True	True	You might want to consider changing to use IBM MQ Client bindings when you move to containers so that you can use smaller containers. Independently scaling the integration servers in your architecture from your queue managers is more easily achieved if you use client bindings rather than server bindings.
II B 2 0	A Healthcare Pack artifact is deployed to this server.	True	True	A Healthcare Pack artifact is deployed to this server. The IBM Integration Bus Healthcare Pack is not supported in IBM App Connect Enterprise 11.0. In IBM App Connect Enterprise v11.0.0.8 (and later), support is provided for applications in healthcare environments through IBM App Connect for Healthcare v5.0.0.0. Consider upgrading to IBM App Connect Enterprise v11.0.0.8 and investigate the features that are provided by App Connect for Healthcare V5.0.0.0, as a replacement for the IBM Integration Bus Healthcare Pack.

Table 7. Rules for the Transformation Advisor tool: (continued)

ID	Title	Can be found in backup file	Can be found in BAR file	Solution
IIB21	A top-level message flow (originating in an Integration project) was found. These artifacts must be moved to the Default Application in IBM App Connect Enterprise v11.	True	True	A top-level message flow (originating in an Integration project) was found. When they are migrated by using the mqsiextractcomponents command, all top-level artifacts are moved to the Default Application in IBM App Connect Enterprise v11. Ensure that you consider the groupings that you require for all top-level resources. The groupings are likely to involve adopting Application projects in preference to Integration projects. Applications and Libraries (which were first introduced in WebSphere Message Broker V8) are encouraged as the preferred way to isolate and group message flows and their associated artifacts. Although BAR files that contain top-level message flows can still be deployed to IBM App Connect Enterprise v11, these artifacts replace previously deployed Default Application content on each deployment. Given this change in iterative deployment behavior, proper consideration of the required groupings of message flows is encouraged.
IIB22	A top-level resource (originating in an Integration project) was found. These artifacts are moved to the Default Application in IBM App Connect Enterprise v11.	True	True	A top-level resource was found. Top-level resources are moved to the Default Application when they are migrated to IBM App Connect Enterprise v11 by using the mqsiextractcomponents command. When you adopt IBM App Connect Enterprise v11, carefully consider your required groupings for all top-level resources. It is likely that the top-level resource that this rule detected is a dependency of a top-level message flow. Consider which message flows depend on this resource. Group the resource so that it continues to be available to the message flow when the flow is moved from its Integration Project to an Application project. You can look for instances of IIB21, which detect top-level message flows, which might have a dependency on the top-level resource that is highlighted by this rule.

Table 7. Rules for the Transformation Advisor tool: (continued)

ID	Title	Can be found in backup file	Can be found in BAR file	Solution
IIB23	A SOAPInput or HTTPInput message flow node was found that is using the integration node-wide listener.	True	False	The message flow contains an instance of a SOAPInput or HTTPInput message flow node that is using the integration node-wide listener. When you run App Connect Enterprise in a container architecture, you do not use an integration node or the integration node-wide listener. Instead you use an independent integration server with its own embedded HTTP listener.
IIB24	Configuration indicates the use of the Record and Replay feature.	True	False	Record and Replay was added to IBM App Connect Enterprise v11 in Fix Pack 11.0.0.4, so it is available if you want to continue to use it. Although you can run this capability in a container-based architecture, its dependence on a relational database and IBM MQ publications might be cause for consideration.
IIB25	A SOAPInput or HTTPInput message flow node that uses HTTPS was found.	True	True	The message flow contains an instance of a SOAPInput or HTTPInput message flow node that is using HTTPS. IBM App Connect Enterprise v11 uses TLSv1.2, and IBM App Connect Enterprise v11.0.0.5 or later can also use TLSv1.3, for inbound HTTPS communications. Ensure that TLSv1.2, or TLSv1.3, are acceptable to your Business Partner applications.
IIB26	A globally coordinated message flow was found.	True	True	A globally coordinated message flow was detected. When you adopt a container-based architecture, you are unlikely to want to use globally coordinated message flows in your containers. You might want to reconsider your architecture to avoid global coordination, or to enable these flows to keep running outside containers.

Table 7. Rules for the Transformation Advisor tool: (continued)

ID	Title	Can be found in backup file	Can be found in BAR file	Solution
II B 2 7	Configuration indicates the use of the embedded global cache feature.	True	False	The embedded global cache feature was added to IBM App Connect Enterprise in Fix Pack 11.0.0.4, so it is available if you want to continue to use it. It is not advisable to use this capability to share information between integration servers in a container-based architecture; if you do, carefully consider the placement and persistence of your catalog servers.
II B 2 8	Configuration indicates the use of the multi-instance high availability feature.	True	False	The multi-instance high availability feature for integration nodes was added to IBM App Connect Enterprise v11 in Fix Pack 11.0.0.3, so it is available if you want to continue to use it. You are unlikely to use this model to achieve high availability if you move to a container-based architecture; if you do, carefully consider the persistence and disk requirements.
II B 2 9	An MRM message set dictionary was detected.	True	True	An MRM message set dictionary was detected. These artifacts are supported for use in IBM App Connect Enterprise v11, but you are advised to consider converting to use the superior DFDL message modeling technology.

Table 7. Rules for the Transformation Advisor tool: (continued)

ID	Title	Can be found in backup file	Can be found in BAR file	Solution
II B 3 0	A message flow with user-defined properties was found.	True	True	A message flow with user-defined properties was detected. Message flows can continue to use user-defined properties when deployed to IBM App Connect Enterprise v11, so this message is for information only. If you are moving to a container architecture, you will probably not want to dynamically change the value of UDPs after deployment. Instead, when you change configuration data, you are likely to prefer to tear down your container and restart it with the new configuration applied. In the unlikely event that you want to dynamically update message flow UDPs after deployment, an administrative API function is available in IBM App Connect Enterprise v11.0.0.6 or later. In general, when you use container-based architectures, other methods for providing configuration to an independent integration server might be preferable, such as a user-defined policy.
II B 3 1	An IBM Integration Bus Activity Log configurable service was detected, which wrote to local files.	True	False	When you migrate to IBM App Connect Enterprise v11, IBM Integration Bus Activity Log configurable services are converted into IBM App Connect Enterprise Activity Log policy documents. When you move to a container-based architecture, you might want to reconsider your chosen output format for IBM App Connect Enterprise Activity Logging.
II B 3 2	An integration server was associated with an integration node that specified a product edition (that uses the mqsimode command) that is no longer available.	True	False	Although artifacts can be carried forward, due to changes in licensing, not all IBM Integration Bus software editions have direct IBM App Connect Enterprise v11 equivalents. Check with your IBM representative to ensure that you move to the appropriate IBM App Connect Enterprise edition and remain licensed correctly in future.

Table 7. Rules for the Transformation Advisor tool: (continued)

ID	Title	Can be found in backup file	Can be found in BAR file	Solution
II B 3 3	An IBM Integration Bus configurable service was detected that cannot be dynamically updated when converted to a policy in IBM App Connect Enterprise v11.	True	False	When you migrate, IBM Integration Bus configurable services are converted into IBM App Connect Enterprise policy documents. IBM App Connect Enterprise policy documents have several advantages, including the fact that they can be created by using Toolkit templates and deployed in a BAR file. Some policies cannot be dynamically updated (without restarting an integration server). This requirement does not usually apply in container-based architectures. However, depending on your use cases, you might want to bear this requirement in mind when you migrate to IBM App Connect Enterprise v11.
II B 3 4	A message flow that uses an MQTT Server was found.	True	True	A message flow uses an MQTT Server, as it contains either an MQTTSubscribe or MQTTPublish message flow node. The built-in MQTT Server that is provided by IBM App Connect Enterprise v11 is not turned on by default within a container.
II B 3 5	A message flow that uses an MQTTSubscribe node to monitor IBM Integration Bus events was found.	True	True	A message flow contains an MQTTSubscribe message flow node with a topic root of "IBM/IntegrationBus". The presence of this node might mean that you have a message flow that is designed to monitor the product itself, and to take some further action when data is received. You might want to revisit this design pattern when you move to a container-based architecture. IBM Integration Bus v10 can be configured to publish to MQTT the following types of information: Operational Events, Admin Events, Business Events, Flow Statistics, and Resource Statistics. IBM App Connect Enterprise v11 can be configured to publish to MQTT the following types of information: BusinessEvents, Flow Statistics, and Resource Statistics. The IBM App Connect Enterprise REST Administration API provides methods that can be called to provide operational and administration information.

Migrating to IBM App Connect Enterprise Version 11.0

Perform the tasks to migrate to IBM App Connect Enterprise Version 11.0.

Before you begin

- Plan your migration; see [“Preparing for migration” on page 1](#)
- Perform pre-migration tasks; see [“Performing pre-migration tasks” on page 22](#)

About this task

Complete the following steps to migrate your deployment to IBM App Connect Enterprise Version 11.0:

- Use extract migration to migrate your integration servers and integration nodes to IBM App Connect Enterprise Version 11.0. You can use extract migration to migrate individual integration servers. Extract migration involves the extraction of configuration and resources from your source system to IBM App Connect Enterprise Version 11.0. Alternatively, you can migrate your integration nodes by using parallel migration:
 - [“Performing extract migration of an integration node or integration server” on page 40](#)
 - [“Performing parallel migration for an integration node” on page 44](#)
- Migrate your development resources to IBM App Connect Enterprise Version 11.0; see [“Migrating development resources to IBM App Connect Enterprise Version 11.0” on page 45](#)

Note: IBM App Connect Enterprise Version 11.0 does not include IBM Integration Explorer and so there is no migration process for IBM Integration Explorer. In IBM App Connect Enterprise Version 11.0, all integration node administration is performed by using the web user interface, the IBM Integration API, or commands. For information about these administration options, see the following topics:

- [web user interface](#)
- [#unique_54](#)
- [Commands](#)

Performing extract migration of an integration node or integration server

You can use the `mqsextractcomponents` command to migrate the configuration and resources of integration servers and integration nodes from your source system to your IBM App Connect Enterprise Version 11.0 system. The command extracts the configuration and resources from your source system and recreates them in a directory structure on your target system. You can extract the configuration and resources from an integration node, including the configuration and resources from all of the integration servers that are associated with the integration node, or from individual integration servers on your source system.

Extract migration of an integration node

The extract migration of the configuration of an integration server or integration node from an earlier version results in the recreation of the integration server or integration node on your IBM App Connect Enterprise Version 11.0 system. A node folder is created in the install directory, or in the directory in which you have your registry. The resources that comprise the integration node configuration on your source system are put into the node folder. The configuration of all the integration servers that are associated with the integration node is also extracted resulting in the recreation of these integration servers on your Version 11.0 system. A server sub-folder is created in the node folder for each integration server. The resources that comprise each integration server configuration on your source system are put into the corresponding server sub-folder.

The following table summarizes the folder structure that is created when you perform extract migration of an integration node.

Table 8.

Folder	Sub-folder	Sub-folder	Sub-folder	Contents/description
<node name>				node.conf.yaml file. This file contains the settings for your migrated integration node. You can override the settings in the node.conf.yaml file by using the mqsichangeproperties command.
	servers			Sub-folders for every integration server that is server associated with the integration node. One sub-folder exists for each integration server.
		<server name>		Sub-folders containing the integration server configuration and resources. The name of the sub-folder is the same as the name of the integration server.
			run	Deployed resources.
			overrides	server.conf.yaml file. This file contains the settings for the integration server. You can override the settings in the server.conf.yaml file by using the mqsichangeproperties command.
	policies			Sub-folders for every policy that is created from a configurable service on the integration node on the source system.
		NodePolicies		
			policy_name.policy.xml	One policy file per migrated configurable service. The file contains the policy information.

Extracting configuration and resources of an integration server

The extract migration of the configuration of an integration server from an earlier version results in the recreation of an independent integration server on your IBM App Connect Enterprise Version 11.0 system. You might want to migrate an integration server in this way if you are operating in a container-based environment. The resources that comprise the integration server configuration are put into a work directory on your Version 11.0 system. You specify the work directory when you use the **mqsextractcomponents** command.

The following table summarizes the directory structure that is created when you perform extract migration of an integration server.

Table 9.

Folder	Sub-folder	Contents/description
<work_directory_name>		server.conf.yaml file. This file contains the settings for your migrated integration server. You can override the settings in the server.conf.yaml file by using the mqsichangeproperties command.
	run	Deployed resources, including policy projects, libraries, and applications.

Table 9. (continued)

Folder	Sub-folder	Contents/description
	overrides	server.conf.yaml file. This file exists if you dynamically override settings in the integration server server.conf.yaml by using the mqsichangeproperties command.
	log	Files containing event log and syslog entries. By default, file names are of the format <code>integration_server.integration_server_name.entries.txt.n</code> where <i>n</i> is an integer in the range 1 through 9. When the integration server generates log entries, the most recent log data is held in a file that does not have the <i>n</i> suffix. As the data ages, it is held in files with the <i>n</i> suffix where the value of <i>n</i> increments with the age of the log data: the oldest data is held in the file that is suffixed 9. The files fill in an incremental fashion; when one file becomes full, entries are written to the next consecutive file.
	config	Files containing other configuration information such as security configuration.

Extract migration of an integration node

You can use the **mqsiextractcomponents** command to migrate the configuration and resources of an integration node, and of all the integration servers that are associated with the integration node, from your source IBM Integration Bus Version 10.0 or IBM Integration Bus Version 9.0 system to IBM App Connect Enterprise Version 11.0.

Before you begin

- Plan your migration; see [“Preparing for migration” on page 1](#).
- Perform any pre-migration tasks; see [“Performing pre-migration tasks” on page 22](#).

Procedure

1. On your source system, run the **mqsibackupbroker** command, specifying the name of the integration node that you intend to migrate to IBM App Connect Enterprise Version 11.0.

The following example backs up integration node INODE on Windows, and saves it in the archive file `C:\MQSI\BACKUP\INODE.zip`:

```
mqsibackupbroker INODE -d C:\MQSI\BACKUP -a C:\MQSI\BACKUP\INODE.zip
```

For more information about the **mqsibackupbroker** command, see [command](#).

2. Transfer the backup file that is created by running the **mqsibackupbroker** command to an appropriate location. You can transfer the backup file to a location on the same system as the existing integration node, or to a location on a different system.
3. Run the **mqsiextractcomponents** command on the system to which you transferred the backup file. On the **mqsiextractcomponents** command, you must specify:
 - The name of the backup file that is created when you run the **mqsibackupbroker** command on the source system.
 - The name in the backup file of the integration node from which you are migrating.
 - The name of the integration node to which to write the extracted configuration and resources.

You can also specify one or more optional parameters on the **mqsiextractcomponents** command. For more information, see [command](#).

4. Optional: If you plan to run the migrated integration node on the same system as the existing integration node, you must stop the existing integration node by running the **mqsistop** command before you start the migrated integration node. For more information, see [command](#).
5. Optional: If you plan to run the migrated integration node on the same system as the existing integration node, and with the same name as the existing integration node, you must delete the existing integration node by running the **mqsdeletebroker** command before you start the migrated integration node. For more information, see [command](#).
6. Start the migrated integration node by using the **mqsistart** command. For more information, see [command](#).
7. After the **mqsistart** command completes, check the syslog, or the Windows Event log to confirm that the integration node has started successfully with no errors.

What to do next

When you complete migration, you might want to complete some post-migration tasks. For more information, see [“Performing post-migration tasks” on page 48](#).

Extract migration of an integration server

You can use the **mqsextractcomponents** command to migrate the configuration and resources of an integration server from your source system to IBM App Connect Enterprise Version 11.0.

Before you begin

- Plan your migration; see [“Preparing for migration” on page 1](#).
- Perform any pre-migration tasks; see [“Performing pre-migration tasks” on page 22](#).

Procedure

1. On your source system, run the **mqsbackupbroker** command, specifying the name of the integration node that is associated with the integration server that you intend to migrate to IBM App Connect Enterprise Version 11.0.
2. Transfer the backup file that is created by running the **mqsbackupbroker** command to an appropriate location on your Version 11.0 system.
3. Run the **mqsextractcomponents** command on your Version 11.0 system.

On the **mqsextractcomponents**, you must specify:

- The name of the backup file that is created when you run the **mqsbackupbroker** command on the source system.
- The name of the integration node with which the integration server that you are migrating is associated.
- The name of the integration server that you are migrating.
- The work directory to which the integration server components are to be extracted.

You can also specify one or more optional parameters on the **mqsextractcomponents** command. For more information, see [command](#).

4. When the **mqsextractcomponents** command has successfully completed, start the integration server on your Version 11.0 system by using the **IntegrationServer** command.

What to do next

When you have completed migration, see the tasks in [“Performing post-migration tasks” on page 48](#) for information about tasks that you might want to complete after migration.

Performing parallel migration for an integration node

You can use parallel migration to perform a staged migration process by creating a new Version 11.0 integration node to run in parallel with your existing integration node. You can then migrate your application logic from your existing integration node to your new integration node.

Before you begin

Complete the following tasks:

- Plan your migration; see [“Preparing for migration” on page 1](#).
- Perform pre-migration tasks; see [“Performing pre-migration tasks” on page 22](#).
- If your deployment uses functions that integrate with IBM MQ (or IBM MQ), or requires that the integration node has a specified queue manager, make sure that you have a supported version of IBM MQ. For information about IBM App Connect Enterprise functions that require IBM MQ, see [Installing](#).
 - If you are reusing an existing IBM MQ (or IBM MQ) deployment, migrate your queue managers and other resources to a supported version of IBM MQ by following the instructions that are provided in the IBM MQ product documentation.

Note: If you are reusing an existing IBM MQ (or IBM MQ) deployment, you can configure the new IBM App Connect Enterprise Version 11.0 integration nodes to use the same queue managers as the existing integration nodes. You do not have to create new queue managers for the new integration nodes.
 - If you are installing a new IBM MQ deployment, create one or more queue managers to support the required configuration for your IBM App Connect Enterprise Version 11.0 integration nodes.
- Check that you have no aggregations in progress on this integration node. When you migrate an integration node to Version 11.0, all live data that is being stored for aggregations in progress is lost.

About this task

To migrate application logic from an integration node on a distributed operating system to a different Version 11.0 integration node on the same computer, or on a different computer, complete the following steps:

Procedure

1. Install IBM App Connect Enterprise Version 11.0 on the same computer as your existing version, or on a different computer.

If you are installing on the same computer, you must specify a different location.
2. Migrate your development resources to the IBM App Connect Enterprise Toolkit Version 11.0; see [“Migrating development resources to IBM App Connect Enterprise Version 11.0” on page 45](#).
3. Set up the correct Version 11.0 command environment:
 - **Linux** On Linux systems, open a new shell and run the environment profile **mqsiprofile** for this Version 11.0 installation.
 - **Windows** On Windows, click **Start**, and open the Command Console that is associated with this Version 11.0 installation.
4. Create a Version 11.0 integration node by using the **mqsicreatebroker** command or the IBM App Connect Enterprise Toolkit. Give the integration node a name that is different from the name of the existing integration node.

If any of the resources on the existing integration node require IBM MQ, specify an appropriately configured queue manager for the new integration node; see [Installing](#).
5. Start the Version 11.0 integration node by using the **mqsistart** command.

You can also start a local integration node by using the IBM App Connect Enterprise Toolkit.
6. Write a list of the integration servers that you have on the existing integration node, and create these same integration servers on the Version 11.0 integration node.

Use the web user interface, the IBM App Connect Enterprise Toolkit Version 11.0, or the **mqsicreateexecutiongroup** command to complete this step.

7. Deploy the message flows, applications, and libraries that are in use by the existing integration node to the Version 11.0 integration node from the Version 11.0 IBM App Connect Enterprise Toolkit.
8. Configure all other relevant properties of the existing integration node on the Version 11.0 integration node.
9. If you want to delete your existing integration node, complete the following steps:
 - a) In a command environment for your previous version, stop the original integration node by using the **mqsistop** command.
 - b) Remove the original integration node from the IBM App Connect Enterprise Toolkit.
 - c) In a command environment for your previous version, delete the original integration node by using the **mqsdeletebroker** command.

What to do next

When you have completed migration, see the tasks in [“Performing post-migration tasks”](#) on page 48 for information about tasks that you might want to complete after migration.

Migrating development resources to IBM App Connect Enterprise Version 11.0

You cannot migrate the IBM Integration Toolkit from previous versions, but you can install IBM App Connect Enterprise Version 11.0 to coexist with a previous version, and migrate the development resources to the IBM Integration Toolkit Version 11.0

Before you begin

Be aware of the following restrictions that apply after you migrate resources to IBM App Connect Enterprise Toolkit Version 11.0.

- You cannot share the development resources with previous versions of the IBM Integration Toolkit. When you create a new project in the IBM App Connect Enterprise Toolkit Version 11.0, it is created in a format that cannot be used in previous versions of the IBM Integration Toolkit. You can take the following steps to manage development with different versions of the IBM Integration Toolkit.
 - If you are using a version control system, create a new stream for use with the new version of your project.
 - If you expect to continue development of a project for an integration node at a version before Version 11.0, you must retain an IBM Integration Toolkit at the previous version.
 - If you need to continue development of the same project for integration nodes on both Version 11.0 and a previous version, ensure you use the IBM Integration Toolkit from the previous version for all your development.
- You cannot deploy resources from IBM App Connect Enterprise Toolkit Version 11.0 to integration nodes at a previous version to Version 11.0.
- If any of the following actions cause an error in the project, the IBM App Connect Enterprise Toolkit flags the error, and you can use a Quick Fix to rectify the error.
 - Creating the metadata information for the user-defined node project
 - Correcting the plug-in identifier if it does not match the project name
 - Ensuring all the user-defined nodes are in the same category

For more information, see [Applying a Quick Fix to a task list error](#). However, if you have different user-defined nodes that depend on different resources that have identical names, the BAR file compiler produces an error that indicates a naming conflict in the dependent resources.

About this task

You can continue to use resources from a previous version of IBM Integration Bus by importing them into an IBM App Connect Enterprise Toolkit Version 11.0 workspace. After you import resources, you can no longer use them in a previous version of the IBM Integration Toolkit.

Message flow projects are replaced by integration projects in IBM Integration Bus Version 9.0 and later. When you import message flow projects into the IBM App Connect Enterprise Toolkit, they are converted automatically to integration projects. You cannot create a message flow project in IBM App Connect Enterprise Toolkit.

To migrate your projects by using a project interchange file, or to convert your message flow projects to integration projects, complete the steps in the following topics:

- [“Importing resources from previous versions” on page 46](#)
- [“Migrating message flow projects” on page 47](#)

Results

You can now view and modify existing resources, and create new resources. You can deploy your workspace resources to IBM App Connect Enterprise Version 11.0 integration nodes.

What to do next

If you migrated from earlier versions, you might want to reorganize your resources by using the containers provided in IBM App Connect Enterprise Version 11.0. Applications are typically used to contain all the resources that are required for a particular solution. Libraries are typically used to contain resources that you might want to reuse. So you might decide to store your main message flow in an application, and store your reusable resources in a library. For instructions about converting your migrated resources to applications and libraries, see [Converting existing projects to applications and libraries](#).

Importing resources from previous versions

You can work with resources from previous versions of IBM Integration Bus by using a project interchange file to import the resources into IBM App Connect Enterprise Version 11.0.

About this task

You might have resources from previous versions of IBM Integration Bus, such as Java projects, and message set projects, that you want to use in IBM App Connect Enterprise Version 11.0. You can import these resources by using a project interchange file.

To import resources into IBM App Connect Enterprise Version 11.0 by using a project interchange file, complete the following steps.

Procedure

1. In the previous version of IBM Integration Toolkit, export the resources that you want to use in IBM App Connect Enterprise Toolkit Version 11.0.
 - a) Click **File > Export**.
The **Export** wizard opens.
 - b) Expand **Other**, click **Project Interchange**, then click **Next**.
 - c) Select the projects that you want to export, and specify the location for the compressed file that is created. You can save the file to a folder on your file system, or to a disk drive.
 - d) Click **Finish**.
The project interchange file is created in the specified location.
2. Create an IBM App Connect Enterprise Toolkit Version 11.0 workspace, or open an existing one.
3. Import the project interchange file into the Version 11.0 workspace.

- a) Click **File > Import**.
The **Import** wizard opens.
- b) Expand **IBM Integration**, click **Project Interchange**, then click **Next**.
- c) Specify the location of the project interchange file that you created in step 1.
- d) Specify the location of the open Version 11.0 workspace.
- e) Select the projects that you want to import into your Version 11.0 workspace, then click **Finish**.

What to do next

You can choose to convert the imported projects to applications or libraries. For detailed instructions for how to convert your resources, see [Converting existing projects to applications and libraries](#).

Migrating message flow projects

When you import resources into an IBM App Connect Enterprise Toolkit Version 11.0 workspace, any message flow projects are converted automatically to integration projects.

Before you begin

Ensure that you imported resources from a previous version of WebSphere Message Broker, as described in [“Importing resources from previous versions”](#) on page 46.

About this task

You cannot create a message flow project in IBM App Connect Enterprise Version 11.0. You might want to continue to use message flow projects if you are working in a team environment, for example.

After your message flow projects are converted to integration projects, the message flows behave in the same way as before. However, if your message flow project contains schema, adapter components, WSDL files, IDL files, or SCA definitions, you might see errors after you migrate your project to IBM App Connect Enterprise Version 11.0.

Quick fixes are available for these errors. These resources have no effect in a message flow project; you can include them in a Message flow project for reference only. But when you include the resources in an application or library, the resources become visible to other applications and libraries. This visibility can affect the resolution of those resources in the workspace. For example, a Message flow project might contain an XSD file that defines an element that is named *element1*, and a message set also contains an XSD file that defines an element that is named *element1*. In previous versions of IBM Integration Bus, this duplication does not cause a problem because the element in the Message flow project is not used for name resolution. However, after migration to IBM App Connect Enterprise Version 11.0, the presence of two elements that are named *element1* causes an error.

An error is also displayed if you import a BAR file with resources from a previous version, and you then choose to refactor those resources to applications and libraries. If you try to recompile a BAR file after the resources are migrated to applications and libraries, you see an error message similar to the following example:

```
TotalPurchaseOrderFlow.msgflow belongs in an application or library and should be deployed within that container and not independently.  
Create a new BAR file and select the application or library in the Prepare tab of the BAR editor, then select Build and Save.  
To deploy the resource separately from the application or library, it must be moved into an integration project.
```

Therefore, you cannot rebuild the BAR file but you can view the contents to see, for example, your original resources and how they were refactored.

What to do next

Decide how you want to organize your migrated resources. Applications are typically used to contain all the resources that are required for a particular solution. Libraries are typically used to contain resources

that you might want to reuse. So you might decide to store your main message flow in an application, and store your reusable resources in a library. For detailed instructions to convert your resources, see [Converting existing projects to applications and libraries](#).

Performing post-migration tasks

After you migrate to IBM App Connect Enterprise Version 11.0, finish setting up your environment.

About this task

Test the IBM App Connect Enterprise Version 11.0 integration node and integration server resources and components to verify that you experience no loss or unexpected change of functionality. Some changes in behavior might be caused by defects that were fixed between versions.

The following topics describe further tasks that you can complete after migration:

Procedure

- [“Reviewing technical changes in IBM App Connect Enterprise Version 11.0” on page 48](#)
- [“Setting up a command environment” on page 49](#)
- [“Reconfiguring administration security after migration” on page 50](#)
- [“Migrating deployable resources” on page 51](#)
- [“Setting up a global cache” on page 50](#)

What to do next

Complete the following tasks as required:

- Delete the components from the previous version.
- Remove the installed code for the previous version.

Access the product documentation for the previous version for details of these tasks.

Reviewing technical changes in IBM App Connect Enterprise Version 11.0

Some minor changes in behavior are present in IBM App Connect Enterprise Version 11.0; for example, those changes that are caused by defects that were fixed between versions.

If you are migrating from Version 9.0, read the following sections to understand the potential effects on your integration nodes and message flows.

- [“Improvements in DFDL schema validation” on page 48](#)

Improvements in DFDL schema validation

The IBM implementation of DFDL has added extra validation checks to comply with the DFDL 1.0 specification. When a DFDL schema that is created in a previous version of IBM App Connect Enterprise or WebSphere Message Broker is validated in the IBM App Connect Enterprise Toolkit or during deployment, errors are reported under the following conditions:

- The DFDL property `length` of an element must not exceed the capacity of the element's simple type. If the validation fails, the following error is generated: CTDV1534E. To prevent the error, change the simple type of the element so that it can cope with the length. For example, if a "two's complement" binary number has the DFDL property `length` set to 8 and the type is `xs:int`, change the type to `xs:long` to clear the error.
- When the DFDL property `initiatedContent` is set to `yes` on the parent sequence or choice, additional checks are made on elements and groups. If the validation fails, one or more of the following errors are generated: CTDV1561E, CTDV1560E, or CTDV1431E. To prevent the error, set

initiatedContent to no on the parent sequence or choice. However, if a new error (CTDV1559E) then appears, contact IBM support for further advice.

- When the parent sequence or choice is the content of a global group, and DFDL properties are placed on group references to the group, additional cross-checks are made between DFDL properties of elements and groups, and DFDL properties on the parent sequence or choice. If the validation fails, one or more of the following errors are generated: CTDV1150E, CTDV1118E, CTDV1432E, CTDV1446E, CTDV1466E, or CTDV1467E. To prevent the error, correct the schema in accordance with the indicated error.
- When the DFDL property occursCountKind of an element is set to parsed, and the parent sequence has a separator, the DFDL property separatorSuppressionPolicy of the parent sequence must be set to anyEmpty. If validation fails, the following error is generated: CTDV1625E. To prevent the error, correct the sequence so that the DFDL property separatorSuppressionPolicy is set to anyEmpty, or change the element so DFDL occursCountKind is set to implicit.
- The DFDL property fillByte must resolve to a single-byte value. If validation fails, the following error is generated: CTDV1458E. To prevent the error, correct the DFDL property fillByte to use a single DFDL entity that resolves to a single-byte value.

Setting up a command environment

Runtime components, and commands that administer runtime components, must be run from a command environment. You must initialize this environment by running the **mqsiprofile** command.

About this task

The **mqsiprofile** command places the Version 11.0 commands and libraries at the front of your search path, and can override any combination of PATH, CLASSPATH, or library PATH.

ODBC settings on Linux systems are found in a text file that is defined by the ODBCINI environment variable. If you are using Linux, set ODBCINI to point to a copy of the sample file *install_dir/server/ODBC/unixodbc/odbc.ini*, where *install_dir* is the IBM App Connect Enterprise installation directory.

On Linux, if you want to use IBM MQ features, you must set the IBM MQ environment where you want the integration server to run; for more information, see [Setting the environment on and](#).

Ensure that you use this environment each time you run an administrative command, or start an integration server.

The following steps explain how to initialize your command environment, either by explicitly running the **mqsiprofile** command (on Linux) or by starting the command console (on Windows):

- On Windows:
 - Open a command console by searching for **IBM Integration Console**. If you have multiple installations of IBM App Connect Enterprise, make sure that you are running the IBM App Connect Enterprise Console from the build of the IBM App Connect Enterprise installation that you want to administer.
- On Linux systems: Locate and run the `mqsiprofile.sh` script in the directory in which you installed the appropriate product.

```
. install_dir/server/bin/mqsiprofile
```

You must include the period and space for this command to work correctly. Add this command to your login profile if you want it to be run at the start of every session.

If you use the zsh shell, running the **mqsiprofile** might cause the terminal session to exit. To resolve this issue, run the **unsetopt function_argzero** command before you run the **mqsiprofile** command.

This command also runs any additional scripts that you copied to the `common\profiles` directory (on Windows) or the `common/profiles` directory (on Linux systems), so that the environment is initialized for runtime components and other resources, such as databases.

What to do next

From the command line in the initialized environment, you can run commands to administer IBM App Connect Enterprise. For example, you can configure and start an integration server, as described in [Configuring an integration server by modifying the server.conf.yaml file](#) and [Starting an integration server](#).

Reconfiguring administration security after migration

After migration, you can reconfigure your administration security to use file-based permissions, queue-based permissions, or LDAP authorization.

About this task

In IBM Integration Bus Version 9.0, you could only configure administration security by using queue-based permissions on IBM MQ queues. In IBM Integration Bus Version 10.0, you could use queue-based permissions or file-based permissions on your integration nodes. In IBM App Connect Enterprise Version 11.0, three modes of authorization are supported for integration nodes and their managed integration servers: file-based permissions, queue-based permissions, or LDAP authorization. For independent integration servers, file-based authorization (*file mode*), and LDAP authorization are supported. For more information, see [Configuring administration security to use file-based, queue-based, or LDAP authorization](#).

In IBM Integration Bus Version 9.0, you used IBM Integration Explorer to administer IBM Integration Bus. In IBM App Connect Enterprise Version 11.0, most of the administration tasks can be completed by using the web user interface. If administrators need to connect to remote integration nodes, you must define web users on the integration node and assign permissions. For more information, see the following topics:

- [Managing web user accounts](#)
- [Controlling access to data and resources in the web user interface](#)

Setting up a global cache

After migration of an integration node that is configured with a global cache, additional steps might be required to complete the global cache configuration.

About this task

In IBM App Connect Enterprise Version 11.0, the global cache configuration involves setting the **cacheServerName** parameter, which must be unique in your global cache system.

If the name of the target integration node is the same as the name of the source integration node, no further configuration is required. However, if the name of the target integration node is different from the name of the source integration node, you must complete the following task to complete the configuration of the global cache:

- Define a unique **cacheServerName** in the `server.conf.yaml` files for the integration servers, and update the `cacheServerName` value in the **catalogClusterEndPoints** property. Consider this example: if the node IB10NODE is migrated from IBM Integration Bus Version 10.0 to IBM App Connect Enterprise Version 11.0 with the name ACE11NODE, the value of **catalogClusterEndPoints** will be `IB10NODE_localhost_2800:localhost:2803:2801`. After migration, specify a value of, for example, `ACE11CatalogServer` in **cacheServerName** and update the value in **catalogClusterEndPoints** to be `ACE11CatalogServer:localhost:2803:2801`, as shown in the following sample global cache stanza for a catalog server:

```
GlobalCache:
  cacheOn: true
  cacheServerName: 'ACE11CatalogServer'
  catalogClusterEndPoints: 'ACE11CatalogServer:localhost:2803:2801'
  catalogDomainName: 'WMB_IB10NODE_localhost_2800'
  catalogServiceEndPoints: 'localhost:2800'
  enableCatalogService: true
  enableContainerService: true
  enableJMX: true
```

```
haManagerPort: 2801
jmxServicePort: 2802
listenerHost: 'localhost'
listenerPort: 2800
```

Similarly, define a unique **cacheServerName** for the container server too. Note that the value of **catalogClusterEndPoints** will be the same for all the integration servers participating in the global cache configuration. The global stanza for a container server, with a **cacheServerName** of `ACE11ContainerServer2` will look like the following example:

```
GlobalCache:
  cacheOn: true
  cacheServerName: 'ACE11ContainerServer2'
  catalogClusterEndPoints: 'ACE11CatalogServer:localhost:2803:2801'
  catalogDomainName: 'WMB_IB10NODE_localhost_2800'
  catalogServiceEndPoints: 'localhost:2800'
  enableCatalogService: false
  enableContainerService: true
  enableJMX: true
  haManagerPort: 2805
  jmxServicePort: 2806
  listenerHost: 'localhost'
  listenerPort: 2804
```

Migrating deployable resources

You can continue to use legacy resources in IBM App Connect Enterprise. However, if you want to continue developing resources created in previous versions, you must migrate them.

Before you begin

- Read the concept information about applications, libraries, and integration projects: [Resource management overview](#).
- Read the concept information about subflows: [Subflows](#).
- You can migrate integration projects by following the instructions in [“Importing resources from previous versions”](#) on page 46.

About this task

You can import integration projects into the Version 11.0 IBM App Connect Enterprise Toolkit to migrate your resources from earlier versions. You can then use an imported project as the basis for a new application or library. Some of your migrated resources need further work before you can continue to develop them.

Note: Message Broker projects were renamed as integration projects in IBM Integration Bus Version 9.0.

The following list outlines the types of resource that require additional migration steps if you want to continue developing them in IBM App Connect Enterprise:

- Message sets: See [“Migrating message sets”](#) on page 52.
- Maps: See [“Migrating message maps”](#) on page 52.
- ESQL files: See [“Migrating ESQL files”](#) on page 53.
- Subflows: See [“Migrating subflows”](#) on page 53.
- Flows that contain JMS nodes: See [“Migrating a flow that contains JMS nodes”](#) on page 53.
- Flows that contain File nodes: See [“Migrating a flow that contains File nodes”](#) on page 54.
- Flows that contain ?wsdl queries: See [“Migrating a flow that supports ?wsdl queries”](#) on page 54.
- Applications that contain SAP adapter projects: See [“Migrating an application that contains an SAP adapter connection project”](#) on page 55

The following table summarizes the type of resource that you must migrate into IBM App Connect Enterprise Version 11.0 if you want to continue developing them:

Table 10. Legacy resources that require additional migration tasks

To continue developing in Version 11.0	Resource created in Version 9.0	Resource created in Version 10.0
Message sets	You must enable the menus for message set development.	You must enable the menus for message set development.
Message maps	You must complete the migration of message maps to graphical data maps for those legacy message maps that you did not convert when you migrated to Version 9.0. New maps created in Version 9.0 are graphical data maps.	You must complete the migration of message maps to graphical data maps for those legacy message maps that you did not convert when you migrated to Version 10.0. New maps created in Version 10.0 are graphical data maps.
ESQL files	You must complete the migration tasks.	No action required.
Message flows	Rework your message flows if they include message flow nodes that are no longer available.	Rework your message flows if they include message flow nodes that are no longer available.
Subflows	You must complete the migration of the legacy subflows that you did not convert when you migrated to Version 9.0. New subflows created in Version 9.0 are created as .subf1ow files.	You must complete the migration of the legacy subflows that you did not convert when you migrated to Version 10.0. New subflows created in Version 10.0 are created as .subf1ow files.

Migrating message sets

About this task

In WebSphere Message Broker Version 8.0 and later, *message model schema* files contained in applications, integration services, and libraries are the preferred way to model messages for most data formats. Message sets are required if you use the MRM or IDOC domains. For more information about message modeling, see [Message modeling concepts](#).

You can import message flows containing message sets from previous versions into IBM App Connect Enterprise Version 11.0. Your existing message sets can be viewed, modified, and deployed in the usual way. However, by default, the menu options for creating new message sets or message definition files are hidden. To perform these tasks, you must first enable message set development in the IBM App Connect Enterprise Toolkit Preferences.

For more information, see [Enabling message set development](#).

Migrating message maps

About this task

IBM App Connect Enterprise Version 11.0 includes a graphical data mapping capability, which is used when you add a Mapping node to a message flow.

You can import message flows containing the following nodes, which use message mapping into IBM App Connect Enterprise Version 11.0:

- DataDelete
- DataInsert
- DataUpdate
- Extract

- Mapping
- Warehouse

The message map (.msgmap) on these Version 7.0 nodes can be viewed but they cannot be deployed to the runtime because they are not available in IBM App Connect Enterprise Version 11.0. The Version 7.0 message mapping operations are accessible only in read-only mode and cannot be modified or deployed.

If you want to deploy a message flow that uses a legacy message map (.msgmap file), you must first replace the legacy node with a new Mapping node, and convert the legacy message map to a graphical data map that can be consumed by the new Mapping node (.map file).

For information about converting Version 7.0 message maps (.msgmap) to Version 11.0 graphical data maps (.map), see [Converting a message map from a .msgmap file to a .map file](#).

Migrating ESQL files

About this task

The required character set encoding for ESQL files that are used in IBM App Connect Enterprise Version 11.0 is UTF-8. When a message flow project is migrated to an integration project, any ESQL files that the message flow project contains are read by using the character set encoding that was used in the previous version of WebSphere Message Broker, and are rewritten in UTF-8. When importing your message flow projects, note that the file encoding of the workspace and host operating system must match the file encoding with which the message flow projects were created.

Migrating message flows

About this task

The set of message flow nodes that are available at IBM App Connect Enterprise Version 11.0 might be different to the set of message flow nodes that are available at the version from which you are migrating. If your message flows contain message flow nodes that are no longer available, you must rework them to use only those message flow nodes that are available at IBM App Connect Enterprise Version 11.0.

Migrating subflows

About this task

Since WebSphere Message Broker Version 8, you create subflows as .subflow files.

Subflows from earlier versions of the product were created as .msgflow files. You must convert them into .subflow files if you want to continue developing them in IBM App Connect Enterprise Version 11.0. You convert these subflows by using the function **Convert to subflow**.

For more information on how to convert a legacy subflow into a .subflow file, see [Converting between message flows and subflows](#).

Migrating a flow that contains JMS nodes

When you migrate a message flow from Version 7.0 that contains JMS nodes, you might have to update the value of the `Transaction mode` property.

About this task

In IBM App Connect Enterprise Version 11.0, the valid values for `Transaction mode` on the JMS nodes are Yes and No. In versions before WebSphere Message Broker Version 8.0, the valid values were Local, Global and None.

When you migrate a message flow that contains JMS nodes with the value of `Transaction mode` set to None, the value is automatically migrated to No.

When you migrate a message flow that contains JMS nodes with the value of `Transaction mode` set to `Local` or `Global`, a migration warning is shown that prompts you to change the value to `Yes`. If you require XA coordinated transactions, you must also select the message flow `Coordinated Transaction` property.

The migration warning is displayed until you change the value appropriately. The warning does not prevent deployment of message flows.

Migrating a flow that contains File nodes

If you migrate message flows that contain File nodes to IBM App Connect Enterprise Version 11.0, ensure that your NFS server appropriately supports file locking.

About this task

If you use an NFS server, and have File nodes in different integration servers in IBM App Connect Enterprise Version 11.0 that access the same directory on the NFS server, ensure that you are using NFS version 4 to correctly support file locking.

Migrating a flow that supports ?wsdl queries

If you want WSDL and XML Schema information to be made available for existing SOAPInput-SOAPReply flows that implement web services, you must explicitly set the SOAPInput node property `Enable support for ?wsdl` and redeploy the flow.

About this task

In the unlikely event you have implemented an HTTPInput-HTTPReply flow to support ?wsdl queries related to a SOAPInput-SOAPReply flow, you should now deprecate the HTTP flow. By sending ?wsdl requests directly to the endpoint exposed by the SOAPInput node, this capability will continue to work even if the internal WSDL deployment details change in the future.

If your HTTP flow and your SOAP flow are using the same listener (either the integration node listener or the same embedded listener), the HTTPInput URL will clash with the SOAPInput URL, and a warning is written to the event log. In this case the ?wsdl request is always serviced correctly, but web service requests could be sent to either the SOAPInput or the HTTPInput node.

Migrating custom integration applications

If you have applications that use the IBM Integration API, check that they access the correct resources.

About this task

Your existing Version 10.0 and Version 9.0 custom integration applications can be migrated to work with Version 11.0 integration nodes. However, if you are writing new applications, ensure that you use the `com.ibm.integration.admin.proxy.*` classes rather than the deprecated `com.ibm.broker.proxy.*`.

The IBM Integration API is no longer asynchronous, because it is backed by synchronous REST calls. As a result, the `AdministeredObjectListener` framework is no longer available, and there are no `processModify()`, `processDelete()` callbacks. You must call `refresh` on the proxy objects to get the latest property values, such as for the deprecated `Execution Group Proxy` `myExecutionGroupProxy.refresh()`.

Procedure

1. If you are migrating from IBM Integration Bus Version 9.0 or earlier, update your custom integration (CMP) applications to use the `IntegrationAPI.jar` file that is included in IBM App Connect Enterprise Version 11.0.

The `IntegrationAPI.jar` file replaces the `ConfigManagerProxy.jar` file that is used in versions before Version 10.0. Existing custom integration applications must be rebuilt to use the `IntegrationAPI.jar` file, and the `IntegrationAPI.jar` file must be included in the class path for the applications. For stand-alone applications, the `IntegrationAPI.jar` file must be supplied with the application.

You might have to change some of the Java calls that the application uses, and you will see red crosses for methods and classes that are no longer available; the Version 11.0 IBM Integration API has removed some `com.ibm.broker.config` classes and methods.

2. For custom integration applications that connect to an integration node, update the connection details that are used by your custom integration applications to connect to the appropriate integration node.

In IBM App Connect Enterprise Version 11.0, the IBM Integration API connections are no longer dependent on IBM MQ, and so connections are no longer made by using IBM MQ queues.

3. If you are connecting to a remote integration node, update your connection details to use the web administration port; for more information, see [Configuring the web user interface](#).

For information about using SSL on these connections, see [Connecting to an integration node by using the Toolkit](#). If you are connecting to a local integration node, you can use a direct local connection.

4. If administration security is enabled, in IBM App Connect Enterprise Version 11.0, user credentials are required for integration applications that use a remote connection. If the connection is local, you must run the application under a user ID that is part of the `mqbkrs` group. For new integration applications, you can provide user details during application development. For existing compiled applications, you must supply the security credentials by using one of the methods that is described in [Connecting to an integration node from a custom integration application](#).

Migrating an application that contains an SAP adapter connection project

If you migrate an application or library that contains an SAP adapter connection project from IBM Integration Bus Version 9.0 to IBM App Connect Enterprise Version 11.0, you must replace the 32-bit version of the SAP libraries with the 64-bit version of the SAP libraries.

About this task

IBM App Connect Enterprise Version 11.0 is only available on 64-bit platforms. If migrate an application that contains an SAP adapter connection project, and the project uses 32-bit libraries, then you must reconfigure the project to use 64-bit libraries.

Note: The 32-bit libraries and the 64-bit libraries have the same file names. If you perform an iterative discovery on an SAP adapter connection project in IBM App Connect Enterprise Version 11.0, and you get a `com.ibm.adapter.framework.BaseException` error, your project is using the 32-bit libraries.

The library file names are detailed in the following list:

- `sapjco3.jar`
- `sapidoc3.jar`
- `sapjco3.dll` (Windows only)
- `libsapjco3.so` (Linux only)

You can configure your SAP adapter connection project to use the 64-bit libraries by using one of the following methods:

- [“Replacing the 32-bit libraries with the 64-bit libraries” on page 55](#)
- [“Updating the Java build path to use the 64-bit libraries” on page 56](#)

Replacing the 32-bit libraries with the 64-bit libraries

Procedure

To replace the 32-bit libraries with 64-bit libraries, complete the following steps:

1. Stop the IBM App Connect Enterprise Toolkit.

2. Navigate to the directory where the 32-bit SAP libraries are stored.
3. Replace the 32-bit SAP libraries with the 64-bit SAP libraries.
4. Restart the IBM App Connect Enterprise Toolkit.

Results

Your SAP adapter connection project is using the 64-bit libraries.

Updating the Java build path to use the 64-bit libraries

Procedure

To update the Java build path for the SAP connection, complete the following steps:

1. In the IBM App Connect Enterprise Toolkit, navigate to the application or library that contains the SAP adapter connection project.
2. Expand **Other Resources** and right-click the SAP adapter connection project; for example, **CWYAP_SAPAdapter_Tx**.
3. Click **Properties** > **Java Build Path** and click the **Libraries** tab.
4. Select **sapidoc3.jar**, click **Edit**, navigate to the location of the 64-bit version of **sapidoc3.jar**, and click **Open**.
5. Select **sapjco3.jar**, click **Edit**, navigate to the location of the 64-bit version of **sapjco3.jar**, and click **Open**.
6. Click **OK** to close the **Properties** dialog.

Results

Your SAP adapter connection project is using the 64-bit libraries.

Index

E

exporting
Project Interchange files [46](#)

I

IBM Integration Bus
Version 10.0
MQ flexibility [8](#)
IBM Integration Toolkit
migration [45](#)
importing
message flow resources [46](#)
Project Interchange files [46](#)

M

message flow projects, migrating [45](#)
message flows
resources
importing [46](#)
migration
?wsdl flows [54](#)
behavioral changes [4](#)
deployable resources [51](#)
File nodes [54](#)
IBM Integration API applications [54](#)
IBM Integration Toolkit resources [45](#)
message flow projects [45](#)
migration tasks [40](#)
preparing [1](#)
reviewing technical changes [48](#)
setting up a command environment [49](#)
setting up a global cache [50](#)
supported paths [2](#)
Version 8.0
JMS nodes [53](#)
MQ flexibility enhancements [8](#)

P

Project Interchange files [46](#)

