WebSphere® IBM Branch Transformation Toolkit for WebSphere Studio

IBM

Version 5.2

Migration guide from version 4.3 to version 5.2

Note!

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 27.

This edition applies to Version 5, Release 2, Modification 0, of *IBM Branch Transformation Toolkit for WebSphere Studio* (5724-H82) and to all subsequent releases and modifications until otherwise indicated in new editions.

IBM welcomes your comments. You can send to the following address:

IBM China Software Development Lab
Branch Transformation Toolkit Product
Diamond Building, ZhongGuanCun Software Park, Dongbeiwang West Road No.8,
ShangDi, Haidian District, Beijing 100094 P. R. China

Include the title and order number of this book, and the page number or topic related to your comment.

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

Contents

Migration guide from version 4.3 to 5.2 . . . 1
Concepts 1
Migration tool architecture. 2
Benefits gained through the migration 3
Migration Overview 4
Migrating your applications to version 5.2 . . . 4
Phase 1: Analysis and preparation 5
Phase 2: Customizing the migration tool . . . 6
Phase 3: Migrating your applications 6
Phase 4: Adding the business logic and fixing errors. 7
Tasks 7
Using the migration tool 7
Setting the migration tool 8
Creating the migration project 8
Migrating toolkit definition files 12
Migrating server operations 17
Migrating flow processors 19
Migrating screen flows processors 20
Generating tooling artifacts 21
Performing a batch migration 22
References 25
Manual works for migration. 25
Diagnosis for the migration tool 26
Notices 27
Trademarks and service marks 29

Migration guide from version 4.3 to 5.2

The migration tool provided by Branch Transformation Toolkit version 5.2 helps users of earlier versions move up to the full power and versatility of the IBM® WebSphere® Application Server and IBM WebSphere Process server. It provides the migration path to allow an enterprise to reuse much of its existing toolkit application code base and still take advantage of the WebSphere J2EE environment.

IBM recognizes that an enterprise cannot always upgrade its application software all at once. Because the enterprise can decide how much or how little toolkit functionality to retain, version 5.2 allows the enterprise to upgrade to a full J2EE environment incrementally. If desired, with version 5.2, the enterprise can bypass toolkit functionality entirely and access J2EE components directly.

This tool is to migrate BTT 4.3 definition to BTT 5.2 base definition. It generates the corresponding program codes and the BTT version 5.2 tooling artifact. There are two ways for the migration:

- Single Migration: You can migrate the BTT 4.3 components one by one manually by using the migration tool. The migration tool will migrate the entities to BTT 5.2 such as dse.ini, data model, formatter, server operations, screen flow, process flow, and generate the artifacts for the tools of BTT 5.2 such as Graphical Builder, BPEL and Struts.
- Batch Migration: You can migrate a batch of components automatically using your own migration application. The batch migration application will migrate the entities and generate the related tooling artifacts. The process can only run on RAD and WID.

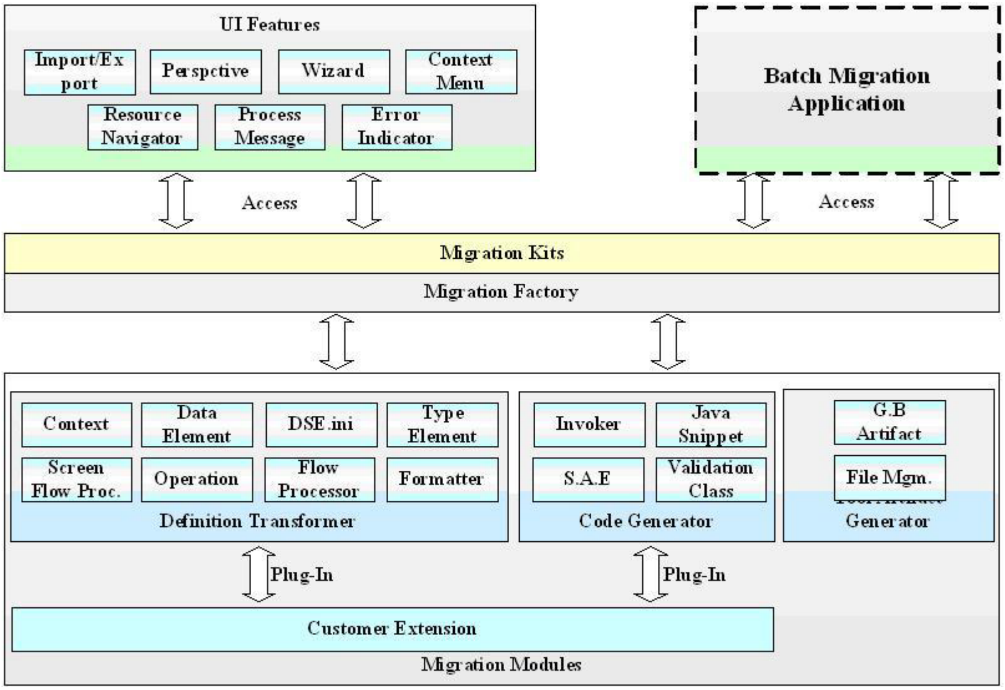
This migration guide describes the migration considerations when you migrate to version 5.2 and the migration tool that helps you roll out the migrations. Note that the descriptions do not present all of the possible architecture variations available in each version.

Concepts

For detailed information about migration from version 4.3 to version 5.2, see the following:

Migration tool architecture

The figure below shows the architecture view of the migration tool.



As shown in the figure above, the migration tool consists of the migration API modules and the UI features plus the custom batch migration application:

- Migration Modules: The migration modules include a set of the functions of the migration process. The categories of the migration modules are:
  - Definition transformer
  - Code generator
  - Tooling artifact generator
  - Custom Extension
- Migration kits: The migration kits are the interfaces between the UI feature (or the batch migration application) and the migration modules. The migration kits access the migration factory to get the correct migration module object, and then invoke the functions of the migration module.
- Migration module factory: It helps the migration kits to instantiate the correct migration module object based on the definition file.
- UI Features: UI features focus on the user interface manipulation. All of the activities that are related to the migration process will delegate to the migration modules through the migration kits.
- Batch migration application: The batch migration application is not a built feature in the migration tool. It has many UI interfaces and it can invoke the migration activities one by one in default order and default parameters.

Import definition file from EAR

There are two ways to input the definition files from an EAR file:

- Flatten the EAR file: This scenario first imports the EAR into RAD, and then all the input files inside the web application WAR file, including definition files and the source code of the application, are extracted and placed in a web application project. Then, the migration tool gets the related files from the specific URL.

Note: This approach will not work if the target files are inside the reference jar files of the web application project. You have to flatten the jar file too in order to get the target files by importing them into the migration tool project or another project, for example, a Java™ application project. Besides, this approach only works on RAD.

- Extract the files from the EAR directly: This scenario extracts the specific files from the EAR file into the related project of the migration directly. In this case, you need to know which package is the target staff from and you need to indicate it for the migration tool. The advantage is that this approach does not have the limitations of the previous one.

To reduce the operation complexity, the second approach is better if the migration tool provides the location list of dse.ini for you to pick up.

Batch migration mode definition file import

To perform the batch migration, the EAR or the dse.ini path needs to be defined in an external file. The migration file can use it as input to get the list.

Handle BTT definition files by BTT Externalizer

To avoid the improper handling of BTT definition, it is better that you use BTT Externalizer to carry out the definition handling and parsing

Benefits gained through the migration

Please help me look at this page, and decide if the benefits listed still apply to 5.2. Thanks.

Branch Transformation Toolkit version 5.2 provides support for distributed architecture. The adoption of a distributed architecture provides some immediate benefits to the system planner and developer:

- Scalability improves by an order of magnitude. A version 5.2 system can scale from a small/medium enterprise having a single server to large enterprises with server farms distributed across multiple regional data centers.
- Performance can be tuned by balancing hardware and network usage against the expected user base. Rational® Application Developer and WebSphere Integration Developer provide many tools and guides for performing this tuning.
- Industry standards such as JCA, EJB 2.0, Struts, and Web services replace many previously architectures and facilities in the toolkit. This adds flexibility to expanding enterprise applications using third-party components built using the same industry standard architectures.
- To enhance its flexibility, the toolkit maintains much of its previous external design philosophy. It further adds more configurability to this mix by allowing its deployment information to be configured. Scaling and performance tuning are all performed using configuration settings rather than Java code changes.
- Platform flexibility is provided using this configurability as well. Version 5.2 can execute using Rational Application Developer for small scale deployments or it can make use of the enhanced features of WebSphere Integration Developer to expand the available options to much larger enterprise deployments.

Read the Solution architecture document for Branch Transformation Toolkit version 5.2 for more information about the benefits coming with version 5.2.

Migration Overview

The following are the components that need to be migrated:

- Input and Output: The input of the migration tool includes BTT 4.3.x definition files from a specific URL or an EAR file, and the source code of the server operations or the validation classes based on BTT 4.3.x framework. The output of the migration tool is the migrated artifacts, such as the related definition files of BTT 5.2, the source code of SAEs, Activities of B.P and the validation classes. The output also includes the tooling artifact for BTT 5.2.
- Definition Transformation: This feature transforms the definition from BTT 4.3 to BTT 5.2. All definition files are converted or migrated to the corresponding BTT 5.2 definition files. For example, screen flow process definition is migrated to Struts base definition, and the business process definition is migrated to BPEL base definition.
- Code Generation: Since many components in BTT 5.2 are different from those in BTT 4.3, the migration tool provides a function to make the prior code migration from the existing BTT 4.3.x program to the BTT 5.2 base program to reduce the workload in the application program migration.
- Tooling Artifact Generation: The migration tool of BTT 5.2 provides a number of development tools to help the developer for the application construction. All the tools have managed artifact which is responsible for the environment or the component extension. So the migration tool can generate the corresponding artifact for each tool. The tooling artifact of BTT 5.2 is based on Rational Application Developer or WebSphere Integration Developer. The migration tool not only includes RAD/WID compatibility, but also the functions. For example, the wsifAction defined in BTT 5.1 tooling artifact is removed when migrated.
- UI Features: The migration provides a simple non-fashion UI to trigger the underlying migration modules to carry out the migration work. It focuses on the management of the user interface and the interaction with the user.
- Customer Extension Support: This is a key feature in the migration tool. It provides the flexibility for you to plug in your own extension. It spans the coverage of the extended functions in the existing BTT application system. You can make the customization to fit the extension and plug it into the migration tool. Then, the tool carries out the customer extension right after the base migration process, or it may totally replace the default process of the migration tool. The extension point is implemented using external definition files and eclipse plugin extension point.
- Modular Base: The migration tool is organized by extension point and external definition file. You can organize the process scenario without worrying about the content processing for the migration.
- Exclusions: The definition file of the communication service and pool services are not covered in the migration tool. You need to migrate them manually
- Constraints: To reduce the complexity, the migration tool limits the availability. You cannot change the migration information, for example, if you change the context name, then the application will not be able to recognize it.

Migrating your applications to version 5.2

The overall migration process consists of the following four main phases:

1. Analysis and preparation

2. Customizing the migration tools
3. Migrating your applications
4. Adding the business logic and fixing errors

The phases are iterative and contain tasks that might be refined during the project life cycle.

Phase 1: Analysis and preparation

Please help decide if this section still applies to 5.2.

During this phase, the project team does a complete analysis on the differences between version 4.3 and version 5.2, especially on the application logic layer. The project team can base on the documentation of the existing application system to identify the part that can be migrated automatically and the part that can not be migrated automatically and prepare what is necessary for the migration. Do the following:

1. Gather all the definition files of the existing application system version 4.3. In the migration tool, it can migration the definition files on the server side from version 4.3 to version 5.2. The migration tool can process the tags based on the specification of version 4.3. If there are any special tag definitions, the migration tool might not be able to migrate it successfully or properly. For this situation, the project team needs to customize the migration tool to extend the process to contain the special definition for the application or do it manually later.
2. The migration tool cannot migrate the business logic of the application to version 5.2 directly. The tool just creates a skeleton with the name corresponding to the server operation or the action in the flow. However, you can make extensions on the migration tool to move the business logic to the generated code automatically. The project team must do an investigation to check whether it is worth or not to extend the migration tool to add more functions.
3. For the integrity, do not customize the migration process on the part of the screen flow generation, the business flow generation and the version 5.2 tooling artifact generation.
4. The project team must package the application into a jar file and then copies the jar file into the plug-in directory of the migration tool and modifies the plug-in.xml to include the library because the migration tool refers to the applications when it does the migration.
5. Because the migration tool can migrate the non-step server operation to a Single Action EJB or a Business Process, the project team must separate the definition files and the application jar into different sets if the application will be migrated to have both set. The migration tool allows you to create multiple projects to migrate different sets of applications.
6. The supported services in the server side leave JDBC Table Service and Electronic Journal Service, and the SNA LU0/6.2 communication services are changed to JCA connectors, the project team must find the substitutive solution (or implement a new service based on the new service architecture) for these unsupported services. For example, Lotus Notes® support, LDAP support, MQ connector, the relevant JCA connector. The migration tool can migrate the server side service definition by modifying the class information regarding the services that supported in BTT 5.2. The migration tool also modifies the DummyDB2Journal.

7. The event mechanism is changed to fit version 5.2 framework to be distributable and span different server. The project team must check and modify the application accordingly if the event mechanism is adopted in the server side in the current application system.
8. For the client application, there is no change. The project team does not need to migrate it to version 5.2. The only thing is to verify the migrating invoker and the related property files to check whether they match the server operation name or not.
9. If the future application only runs on WebSphere Application Server, use Rational Application Developer to migrate the application. If the future application runs on WebSphere Process Server, using WebSphere Integration Developer and you will have more features supported. The migration tool will provide fewer features on Rational Application Developer than WebSphere Integration Developer. The difference is Rational Application Developer will not support generation of the Business Process. In Rational Application Developer, the step operations or the flow processor will not be able to migrate. The project team needs to change it to be non-step operations, or just skip it and create the Single Action EJB manually. Also, the non-step operation can only be migrated to a Single Action EJB.

   Related information

   Events

   Bean Invoker Factory

Phase 2: Customizing the migration tool

During this phase, the project team can customize the migration tool to enrich the functionality. To do this, the project team needs to do the following:

1. Customize the code generation process. Because the migration tool only generates a skeleton when the tool does the migration, the project team can enhance the functionality. For example, copy the business logic from the server operation or the action of the flow to the code generation logic by extending the classes of the migration tool and changing.
2. Customize the definition file migration process by extending the migration tool base classes and changing.
3. Customize the user interface to the custom migration process. The user interfaces of the migration tool are based on Eclipse programming model.

Phase 3: Migrating your applications

During this phase, the project team can start to migrate the application by using the migration tool. To do this, the project team needs to do the following:

1. Create a migration project by using the migration tool and import the definition files from a specific EAR file, or specify the dse.ini file in the disk. After the process finished, the migration tool will create the related projects, like EAR project, Web project, EJB project and Services Project if WebSphere Integration Developer is adopted.
2. Migrate dse.ini file.
3. Migrate the definition files of the context, the data element, the type data and the format.
4. Migrate the business logic:
   - Migrate the server operations to Single Action EJBs or business processes if WebSphere Integration Developer is adopted. After you complete the process, the migration tool will generate things like invoker (Java code), and Single Action EJB or business process.

- Migrate the flow processes to business process if WebSphere Integration Developer is adopted. After you complete the process, the migration tool will generate the things like BPEL file, invoker (Java code).
- Migrate the screen flow to Struts Extensions. After you complete the process, the migration tool will generate things like struts configure file, web diagram (.gph file) and migrated JSP.
- Migrate self-definition files if there are any. The output includes the above 3 output files. Also, the definitions of the context, the data element, the type data and the format in the self-definition files will be add into the generic definition files.

Note: Business processes do not automatically chain process contexts (equivalent to operation contexts and flow processor contexts in version4.3) to session contexts. You need to chain the process contexts to session contexts manually.

5. Generate version 5.2 tooling artifact. The migration tool can generate the artifact of Graphical Builder based on the generation of the screen flow.

Phase 4: Adding the business logic and fixing errors

After the project team has migrated the application from version 4.3 to version 5.2, the project team must add the business logic and fix errors. The project team needs to do the following:

1. Add the business logic by moving the code from the source code of the current application if the project team does not enhance the migration tool to have the capability to copy the business logic code into the generated code.
2. Modify the code about the package name, the access to the context, the data elements, the format, and the JDBC services and so on.
3. Build the services that version 5.2 does not support. Change the access to the services.
4. Change the access to the communication services to comply with JCA.

Tasks

For the tasks related to migration from version 4.3 to version 5.2, see the following:

Using the migration tool

Branch Transformation Toolkit provides a set of tools to help you migrate your toolkit applications developed with version 4.3 of the toolkit to the new version 5.2 architecture.

The migration tool does the following things for you during a migration:

- Migrate version 4.3 definition files to version 5.2 definition files, including dse.ini, context definitions, formatter definitions, and so on
- Generate the corresponding codes, for example, invoker classes for the Java client presentation handler, S.A.E or Java snippet for the server operation, and so on
- Generate version 5.2 tooling artifact such as Graphical Builder definition files

The migration tool provides a customer extension feature to help you plug-in the custom process into the migration module. You can base on the given migration module to make the customizations to fit the extension and plug it into the migration tool. Then the migration tool carries out the customer extensions after the base migration process, or it can totally replace the default process of the migration tool.

The migration tool provides graphical user interfaces (GUIs) to help you migrate the toolkit configuration files, data model, formatter definitions, server operations, flow processes, and screen flows of your version 4.3 application to the corresponding components of your version 5.2 application. Furthermore, the migration tools are designed to be a scalable toolset, so that you can build your own batch migration applications using a set of APIs. You can also customize the migration tools for your special needs.

Note: After migration, you should not change the names of the toolkit constructs such as contexts, formatters, and so on. The names for the constructs are assigned by the migration tools automatically. If you change the name of a construct manually after the migration, the migrated application will not be able to find that construct anymore.

Setting the migration tool

Before you migrate your application by using the migration tool, you must do the following things:

1. Package the server side application into jar files.
2. Copy the jar files into the migration tool plug-in directory under runtime folder.
3. Modify plug-in.xml to include the jars as the runtime library. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin
id="com.ibm.btt.tools.migration"
name="BTT Migration Tool Plug-in"
version="1.0.0"
provider-name="IBM"
class="com.ibm.btt.tools.migration.BTTMigrationPlugin">

<runtime>
    <library name="bttmigration.jar">
        <export name="*"/>
    </library>

    <library name="runtime/bttstrutstool.jar"/>

    <library name="runtime/dseb.jar"/>
    <library name="runtime/dseflp.jar"/>
                          :
                          :
    <library name="runtime/yourapp1.jar"/>
    <library name="runtime/yourapp2.jar"/>
</runtime>
```

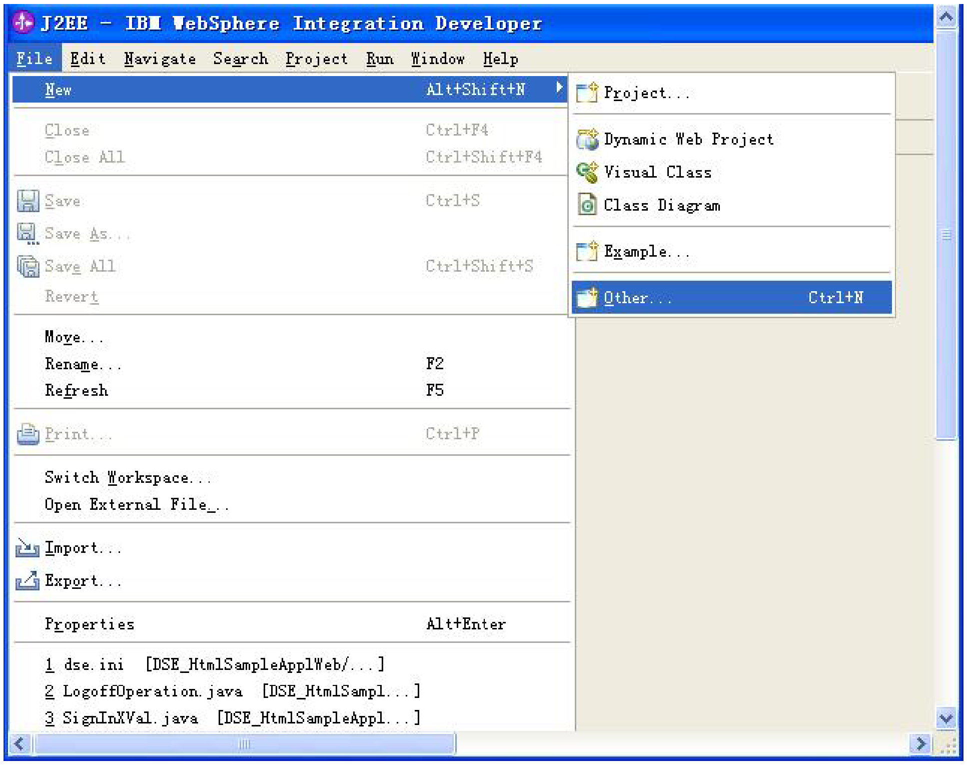
These steps are necessary for the migration because the migration tool must use the jar files to initialize BTT environment and read the related information about your application.

Creating the migration project

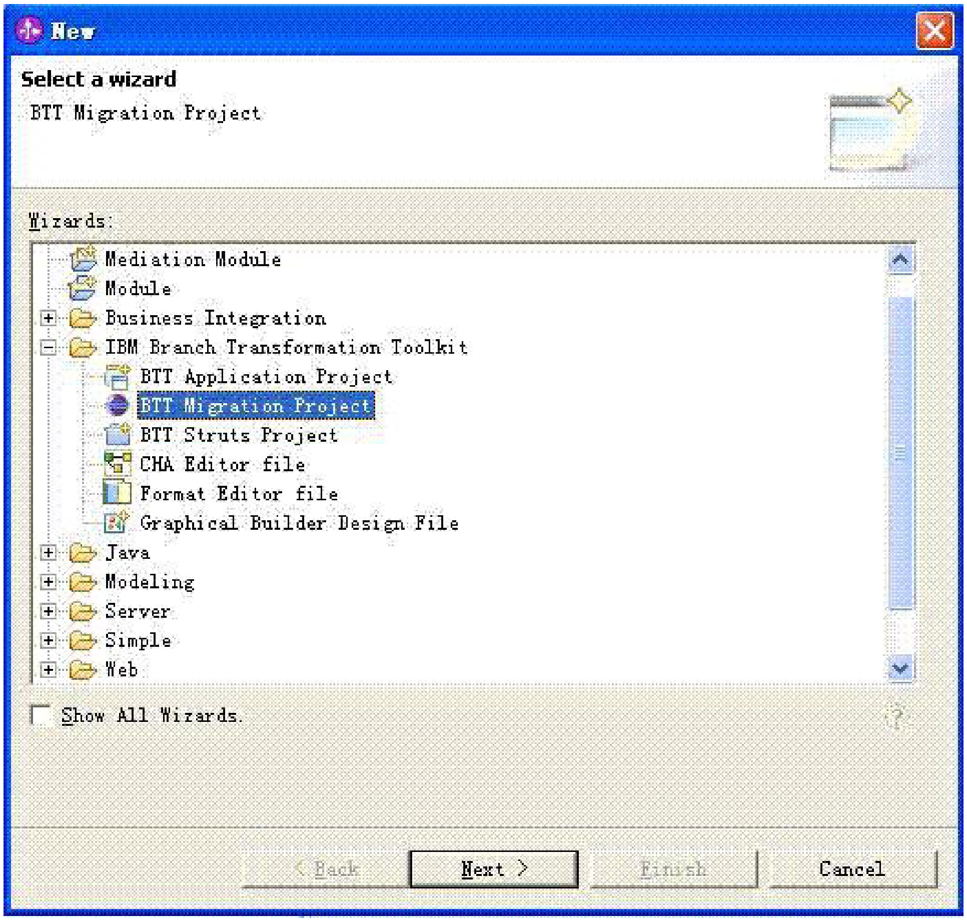
After you have done the migration analysis and customized the migration tool for your special needs, you start doing the migration by creating an migration project.

To create an migration project, do the following:

1. From the WebSphere Integration Developer menu, select **File** → **New** → **Other**.



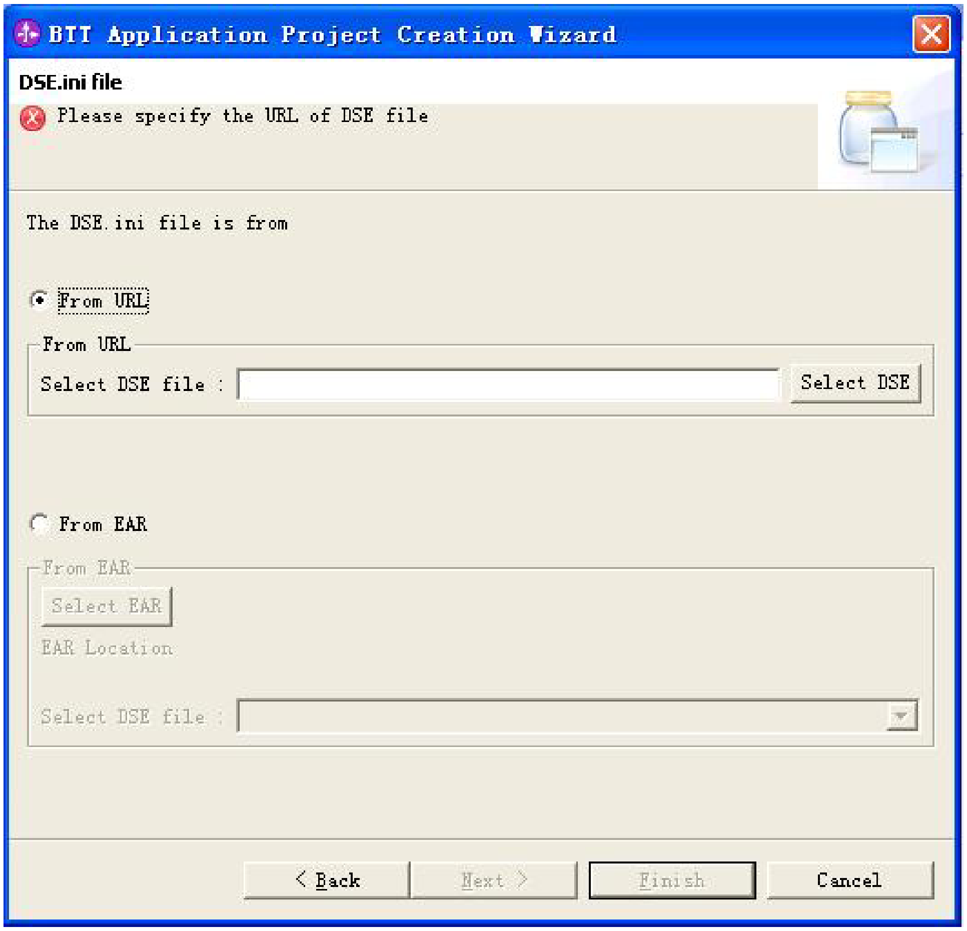
2. In the dialog box that pops up, expand **Branch Transformation Toolkit**, and then select **BTT Migration Project**. Click **Next**.



Select a wizard

BTT Migration Project

3. From the Dse.ini panel, you select the dse.ini file of the project you are migrating.



BTT Application Project Creation Wizard

DSE.ini file

Please specify the URL of DSE file

The DSE.ini file is from

● From URL

○ From EAR

You can choose the dse.ini file in the following ways:

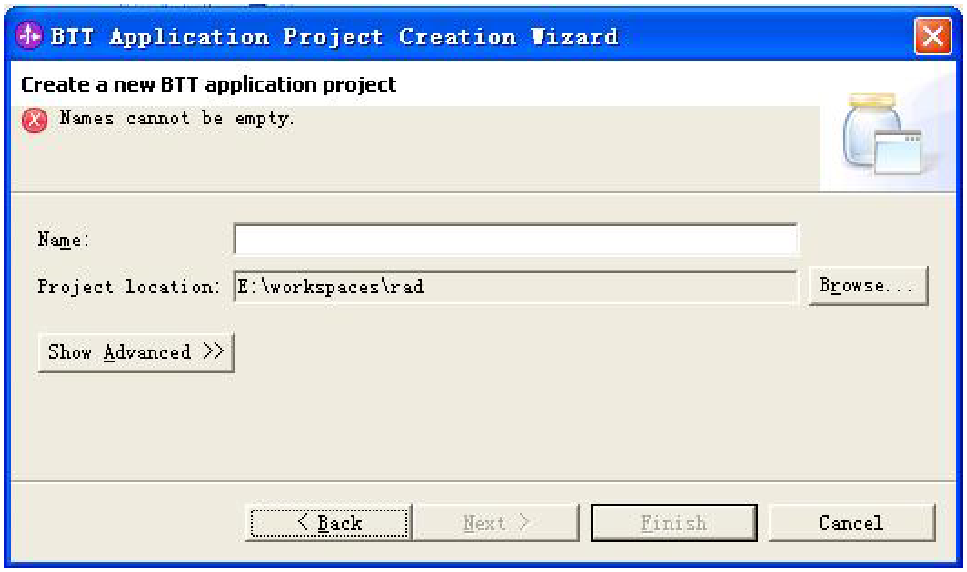
- By selecting the dse.ini from your local directory.

  To do this, you select the **From URL** option, and click the **Select DSE** button to browse to your dse.ini file and select it.
- By selecting the EAR file that contains the dse.ini file.

  To do this, you select the **From EAR** option, and click the **Select EAR** button to browse to your EAR file and select it. You can then select the dse.ini file from the **Select DSE File** list.

Note: The migration tool only migrates server-side application components, so make sure you the dse.ini file you selected is the server-side one.

4. Click **Next**.
5. Enter the name of your migration project and select the project location. Click **Next**.



BTT Application Project Creation Wizard

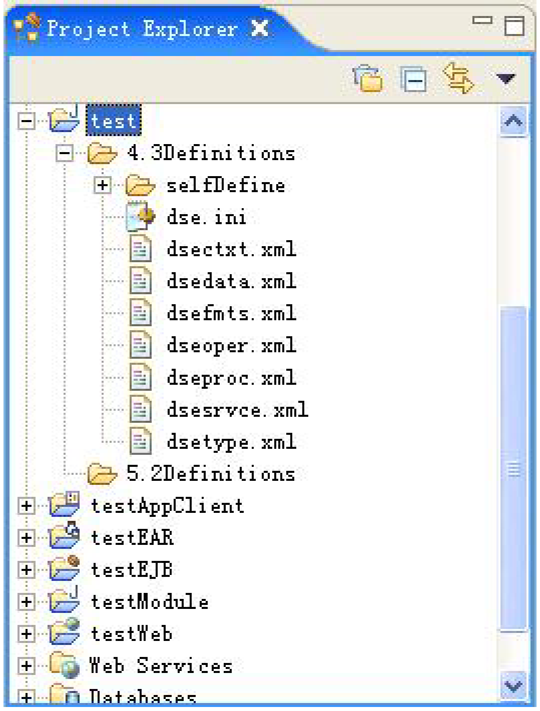
Create a new BTT application project

Names cannot be empty.

Name:

Project location: E:\workspaces\rad

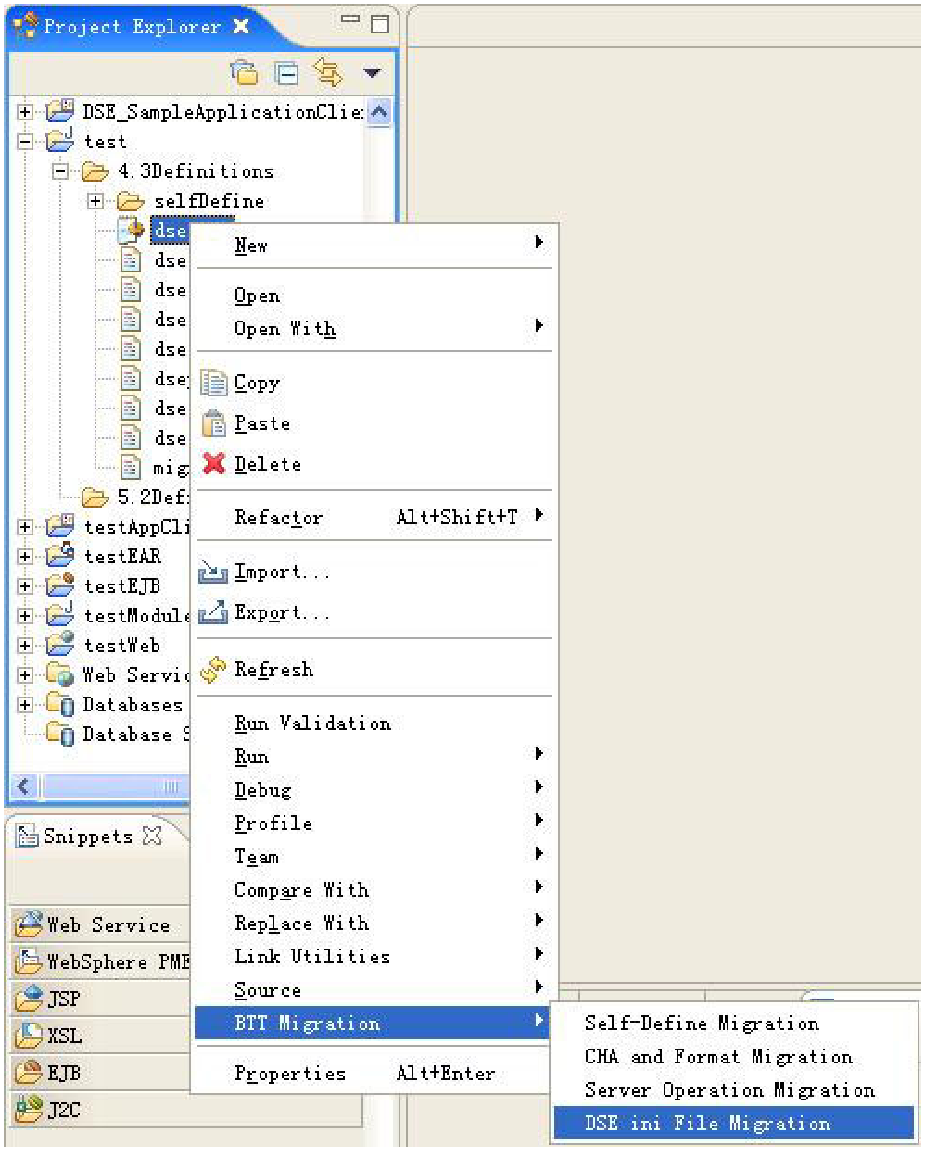
After you create the migration project, you can find all related 4.3 definition file are copied. and the projects of EAR, EJB, Web and Services are also created.



Migrating toolkit definition files

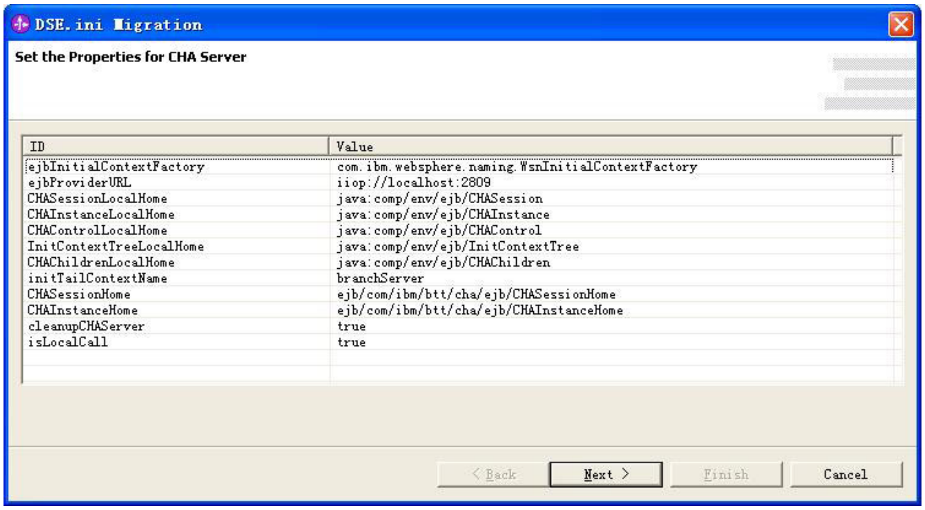
To migrate the version 4.3 toolkit definition files to version 5.2 definition files, do the following:

1. Migrate the toolkit configuration file, dse.ini:
   a. From the Package Explorer view of your WebSphere Integration Developer, right-click the dse.ini file that you want to migrate, and select **BTT Migration** → **DSE INI File Migration**.



The dse.ini file is in the 4.3 definitions folder of your application migration project. This starts the dse.ini file migration wizard.

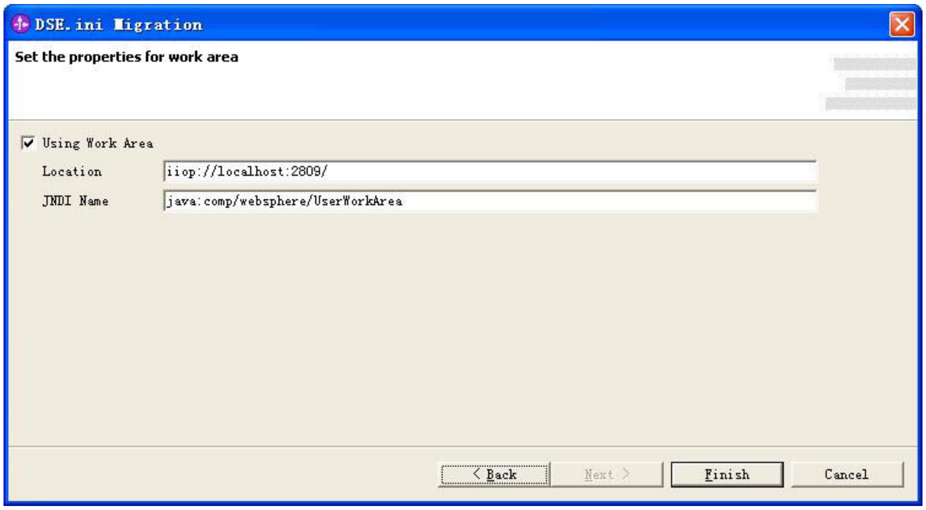
b. Follow the wizard, set the properties for the CHA server:



Set the Properties for CHA Server

ID	Value
ejbInitialContextFactory	com.ibm.websphere.naming.WsnInitialContextFactory
ejbProviderURL	iiop://localhost:2809
CHASessionLocalHome	java:comp/env/ejb/CHASession
CHAInstanceLocalHome	java:comp/env/ejb/CHAInstance
CHAControlLocalHome	java:comp/env/ejb/CHAControl
InitContextTreeLocalHome	java:comp/env/ejb/InitContextTree
CHAChildrenLocalHome	java:comp/env/ejb/CHAChildren
initTailContextName	branchServer
CHASessionHome	ejb/com/ibm/btt/cha/ejb/CHASessionHome
CHAInstanceHome	ejb/com/ibm/btt/cha/ejb/CHAInstanceHome
cleanupCHAServer	true
isLocalCall	true

c. Set the properties for the work area.



Set the properties for work area

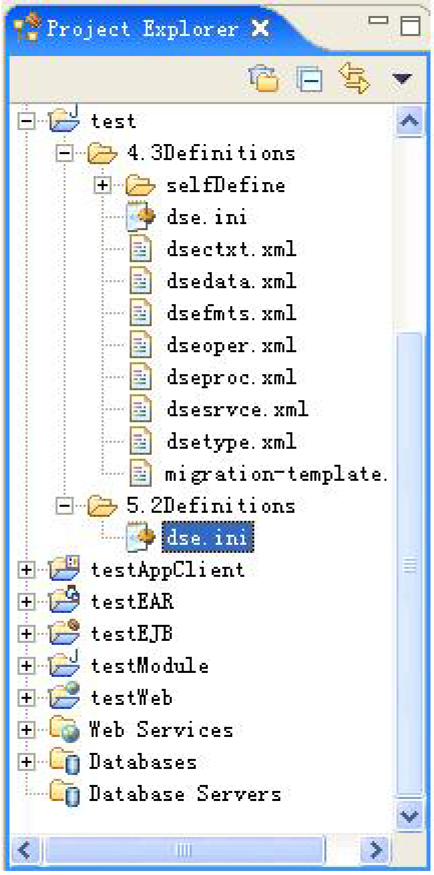
☑ Using Work Area

Location: iiop://localhost:2809/

JNDI Name: java:comp/websphere/UserWorkArea

d. Click **Finish**.

The wizard migrates your version 4.3 dse.ini file to a version 5.2 one, and copies the 5.2 dse.ini file into the 5.2 definitions folder of your application migration project.



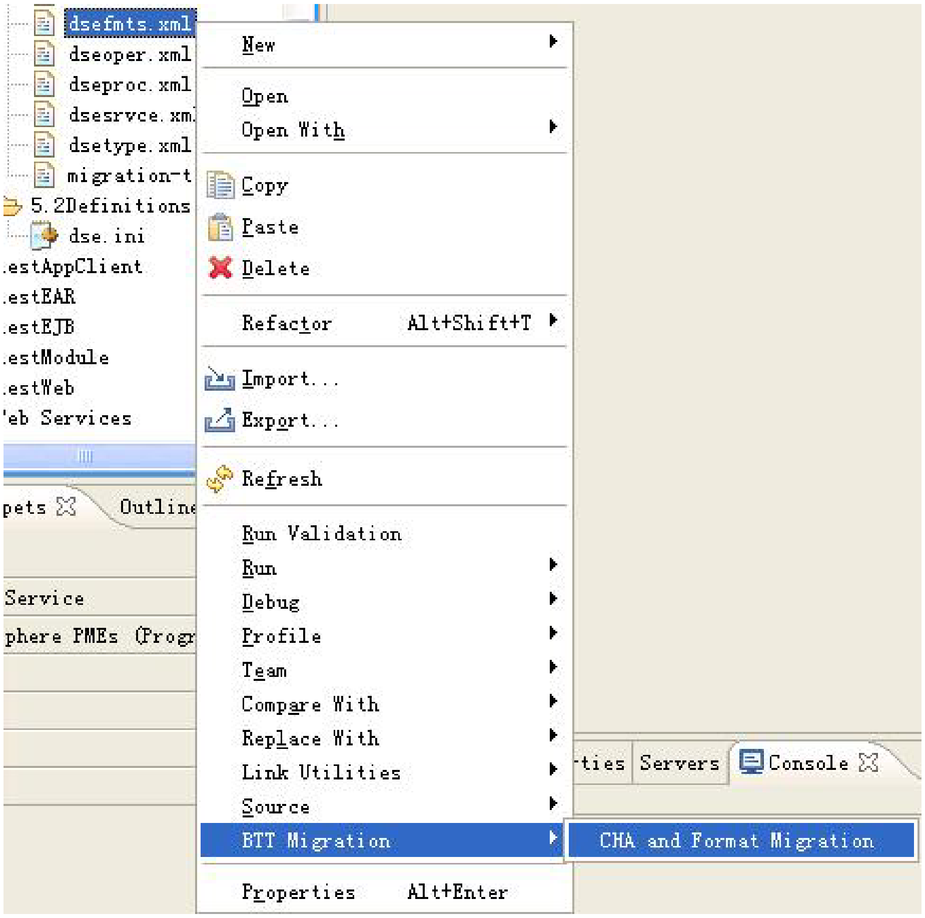
2. Migrate the definition files:
   a. Making sure that the IDs of your data field start with lowercase characters. If the first character of a data field ID is an uppercase character, change that first character into a lowercase one. Here's an example:

```
<kColl id="individualSessionData">
    <field id="LogonPortal" value="1"/>
</kColl>
```

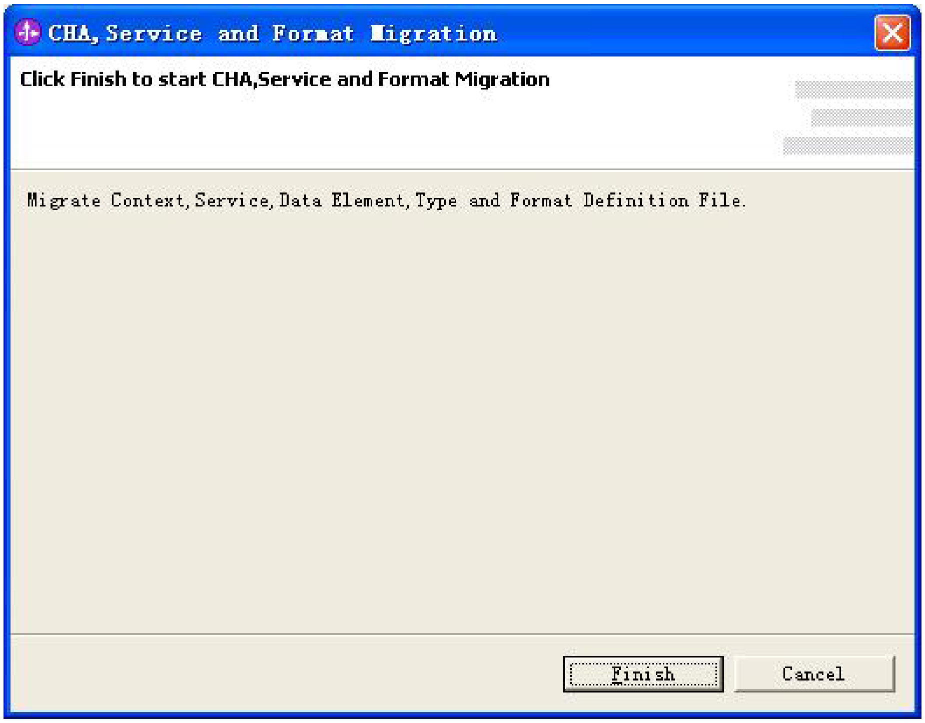
   You need to change the preceding codes to the following ones:

```
<kColl id="individualSessionData">
    <field id="logonPortal" value="1"/>
</kColl>
```

   b. From the Package Explorer view of your WebSphere Integration Developer, right-click the your version 4.3 dsefmts.xml file, and select **BTT Migration** → **CHA and Format Migration**.



The dsefmts.xml file is in the 4.3 definitions folder of your application migration project. Then click **Finish** to start CHA, service and format migration.



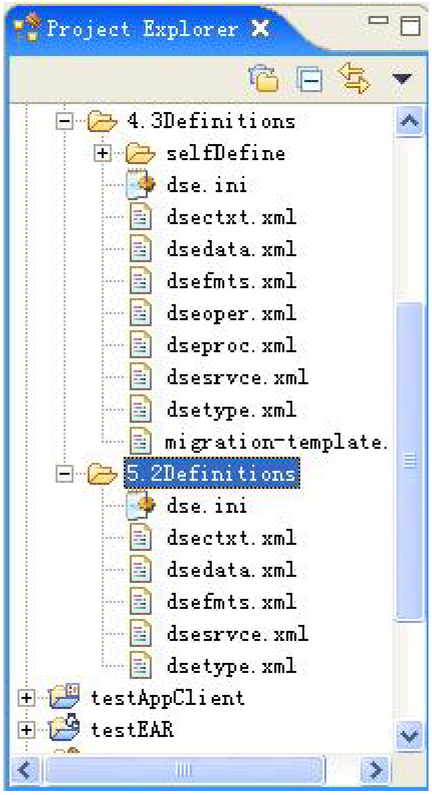
Click Finish to start CHA,Service and Format Migration

Migrate Context, Service, Data Element, Type and Format Definition File.

The migration tool migrates the following files from version 4.3 to version 5.2:

- dsedata.xml
- dsetype.xml
- dsectxt.xml
- dsefmts.xml
- dsesrvce.xml

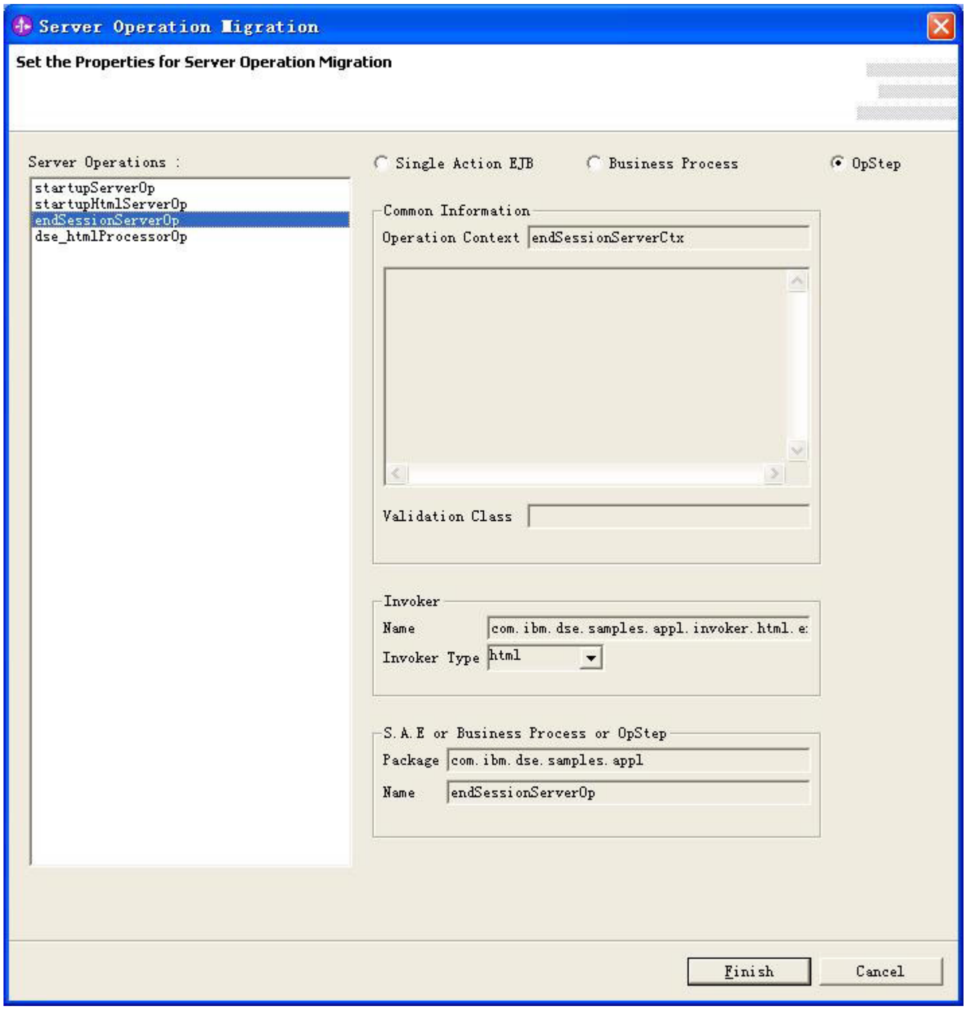
The migrated files are copied to the 5.2 definitions folder of your application migration project.



Migrating server operations

To migrate your 4.3 server operations to Single Action EJBs, or Business Processes, or server operations in 5.2, do the following:

1. From the Package Explorer view of your WebSphere Integration Developer, right-click the your version 4.3 dseoper.xml file, and select **BTT Migration** → **Server Operation Migration**. This starts the server operation file migration wizard.


Server Operation Migration
Set the Properties for Server Operation Migration
Finish
Cancel

Server Operations :
- startupServerOp
- startupHtmlServerOp
- endSessionServerOp
- dse_htmlProcessorOp

○ Single Action EJB ○ Business Process ● OpStep

Common Information
Operation Context: endSessionServerCtx
Validation Class:

Invoker
Name: com.ibm.dse.samples.appl.invoker.html.e:
Invoker Type: html

S.A.E or Business Process or OpStep
Package: com.ibm.dse.samples.appl
Name: endSessionServerOp

2. From the Server Operation Migration dialog box shown in the figure above, select the server operations.

   An invoker will also be generated for calling the Single Action EJB or business process or operation steps. From the Server Operation Migration dialog box, you can see the common information, and the name and type of the invoker, and the name of the Single Action EJB or Business Process or operation steps, and also the package where they locate

3. Click **Finish**. You can then see the generated invoker and its properties files in the Web project of your application, the Single Action EJBs in the EJB project of your application, or the business processes in the Process project of your application, or register information for operation steps. The entry properties file is also upgraded in the application projects.

   Note: The entry properties file is a kind of index or an invoker registry to map an invoker ID to an invoker properties file.

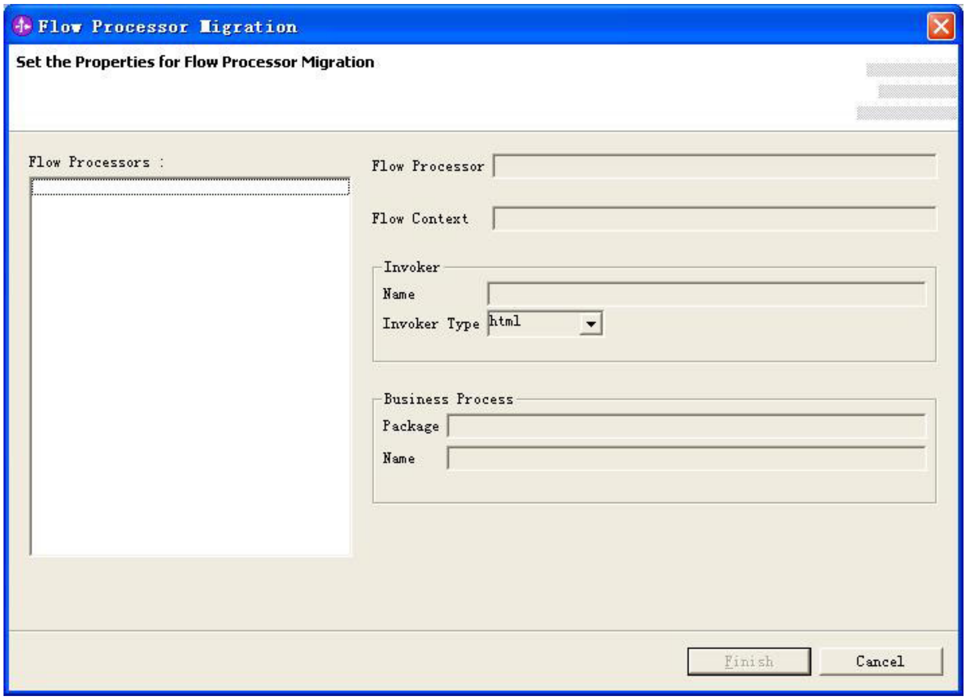
   The operation steps are migrated to Java snippets for business processes. Links between the Java snippets are generally automatically based on the server operation definition files.

   If the operation contains sub-operations, the stepped sub-operations will be migrated to business processes with associated invokers generated, and the non-step sub-operations can be migrated to Java snippets of the BaseOperSnippet type.

Migrating flow processors

In Branch Transformation Toolkit version 5.2, flow processors are replaced by business processes on the server side. To migrate your server-side flow processors to the corresponding constructs in version 5.2, do the following:

1. From the Package Explorer view of your WebSphere Integration Developer, right-click the your dseproc.xml file that contain the version 4.3 flow processors, and select **BTT Migration** → **Flow Processor Migration**. This starts the flow processor file migration wizard.



Flow Processor Migration

Set the Properties for Flow Processor Migration

Invoker Type html

2. From the Flow Processor Migration dialog box that pops up, select the flow processors.
   - The Actions defined in your flow processor definition file maps to Java snippets.
   - Guard conditions map to extra navigation links, as a leaf node in the business process. Conditions are added in links to control the navigation.
   - Transition conditions map to normal business process nodes.

   From the Flow Processor Migration dialog box , you can see the name and type of the invoker, the name of the business process, and the package where the business process locates.
3. Click **Finish**.

Java snippets and business processes are created in the Process project of your application. Validation classes are also migrated.

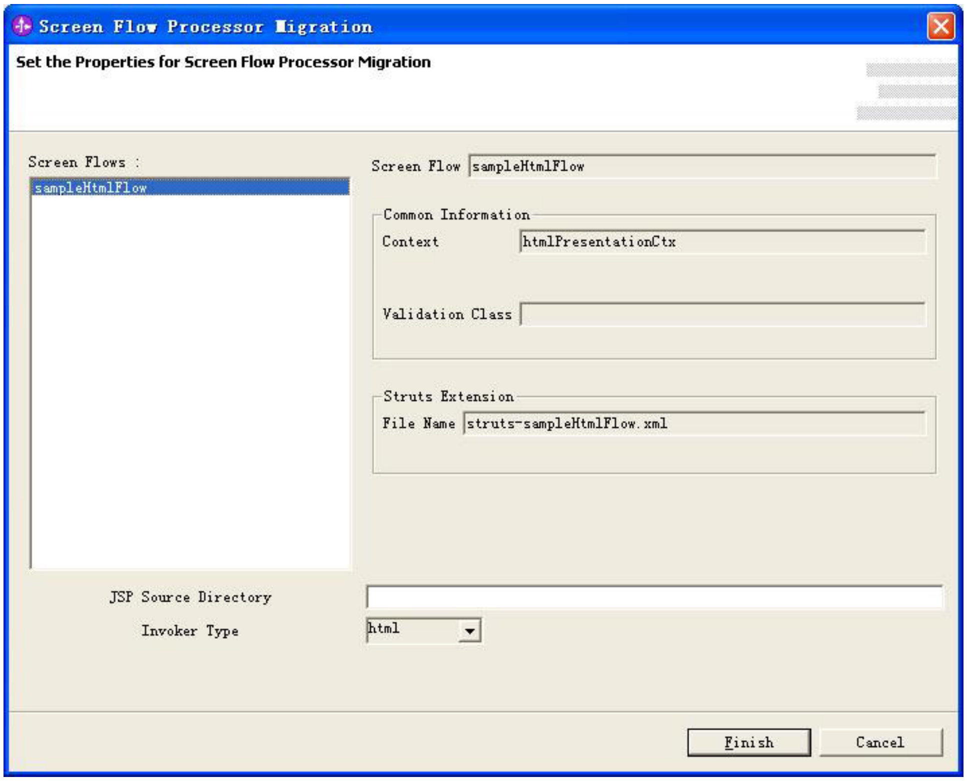
Note: Make sure you meeting the following requirements when migrating flow processors:

1. Flow processor do not contain any page state.
2. The sub-processors or descendent processors of the flow processor that you are migrating do not contain any page state.
3. The flow processor definition file follows all the rules for defining a processor.

Migrating screen flows processors

In Branch Transformation Toolkit version 5.2, screen flow processors are re-based on the Apache Struts Framework. To migrate your version 4.3 screen flows to the corresponding constructs in version 5.2, do the following:

1. From the Package Explorer view of your WebSphere Integration Developer, right-click the your dseproc.xml file that contains your version 4.3 screen flow processors, and select **BTT Migration** → **Screen Flow Processor Migration**. This starts the screen flow processor file migration wizard.



Screen Flow Processor Migration

Set the Properties for Screen Flow Processor Migration

Screen Flows :
sampleHtmlFlow

Screen Flow: sampleHtmlFlow

Common Information
Context: htmlPresentationCtx
Validation Class:

Struts Extension
File Name: struts-sampleHtmlFlow.xml

JSP Source Directory:
Invoker Type: html

2. From the Screen Flow Processor Migration dialog box that opens, select the screen flow processors. From the Screen Flow Processor Migration dialog box , you can see the common information, like context, validation class, and Struts file name.
3. Enter the **JSP Source Directory**, and select the **Invoker Type**.
4. Click **Finish**.

The migration of screen flow processors does the following for you:

- Convert the <htmlProcessor> tag to a Struts process definition file.
- Convert Actions to Struts Actions.
- Convert operation actions to business processes, defined in a BPEL file.
- Translate the navigation links into the Struts definition file. Transitions are transformed into the forward definitions of Struts.
- Convert version 4.3 validation classes to version 5.2 validation classes.
- Convert condition definitions to the condition definitions of Struts Extensions.
- Convert processor contexts to the processor CHA contexts of Struts Extensions.
- Convert transition actions to two Struts actions.
- Convert transition contexts to transition CHA contexts and Form Beans of Struts Extensions.

- For JSP files, change the *next event* to a URI.

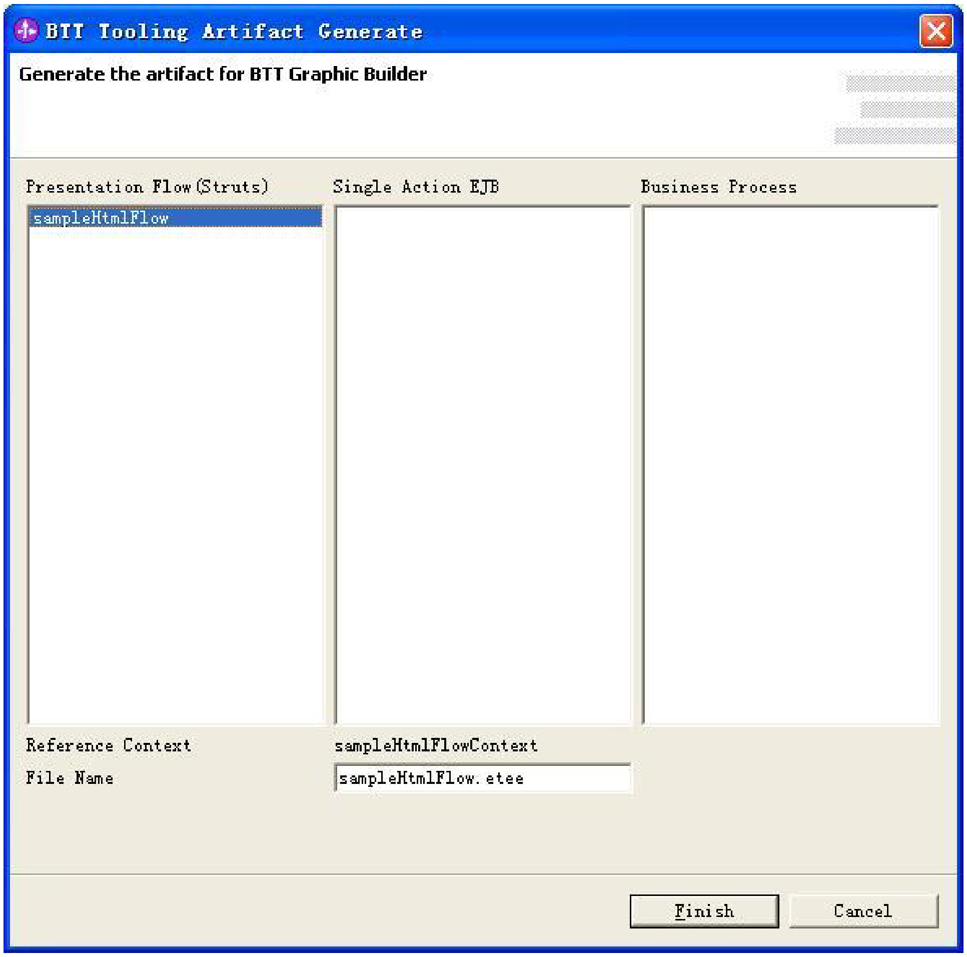
After the screen flow migration process finishes, the result will be placed in the web project. You can browse and edit the screen flow by the Struts configuration file in an XML editor or the .gph file in a graphical screen flow editor.

Generating tooling artifacts

Apart from migrating your version 4.3 definitions and runtime components to version 5.2 constructs, the migration tool can also generate artifacts that you can use with the toolkit version 5.2 tools to further develop or maintain your application. This function can generate a Graphical Builder file for you, and when you open the file with Graphical Builder, you'll see the components and definitions you migrated in the Graphical Builder views.

To generate tooling artifacts with the migration tool, do the following:

1. From the Package Explorer view of the WebSphere Integration Developer, right-click your migration project and select **BTT Migrate** → **BTT Tooling Artifact Generate**.
2. From the BTT Tooling Artifact Generate dialog box that opens, select the screen flows, business processes, and Single Action EJBs from which you want to generate the tooling artifacts.



BTT Tooling Artifact Generate

Generate the artifact for BTT Graphic Builder

Presentation Flow(Struts): sampleHtmlFlow

Single Action EJB

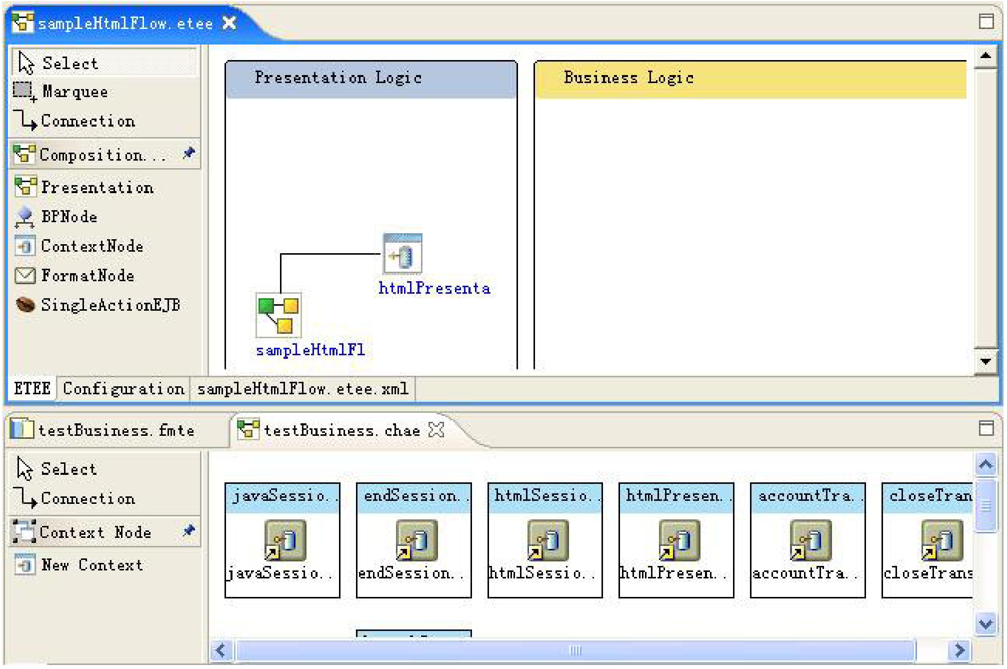
Business Process

Reference Context: sampleHtmlFlowContext

File Name: sampleHtmlFlow.etee

3. Click **Finish**.

You can then find the Graphical Builder file (.eete) file in your migration project. You can then double-click the Graphical Builder file to start the Graphical Builder, and you can see the components you selected for artifacts generation are there represented by Graphical Builder nodes.



Related information

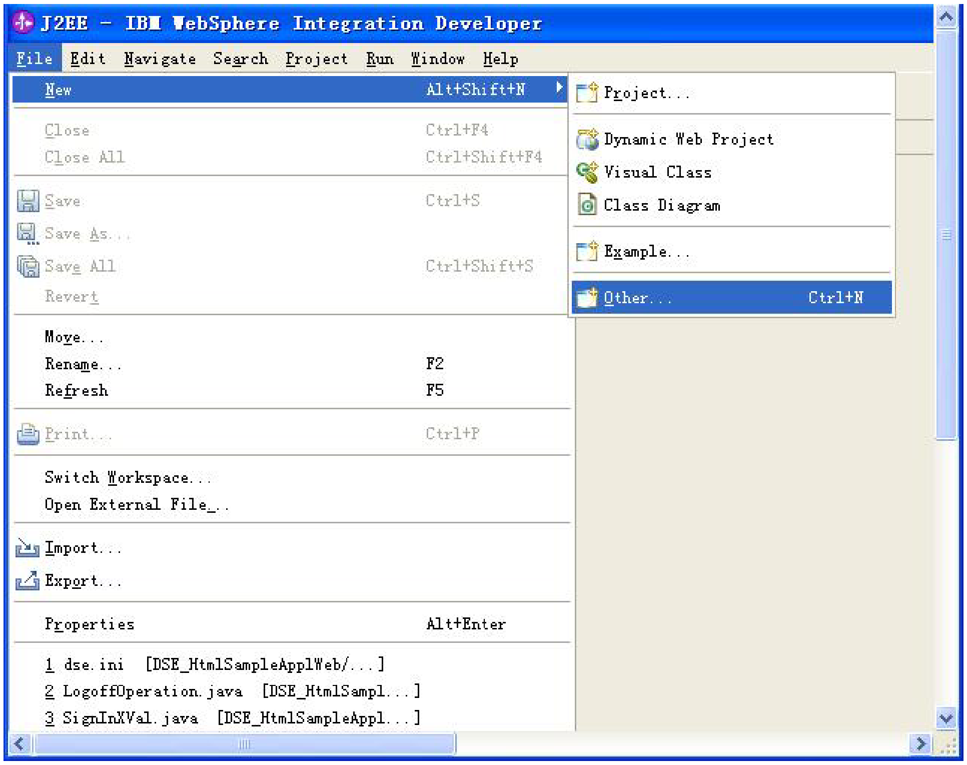
Graphical Builder

Performing a batch migration

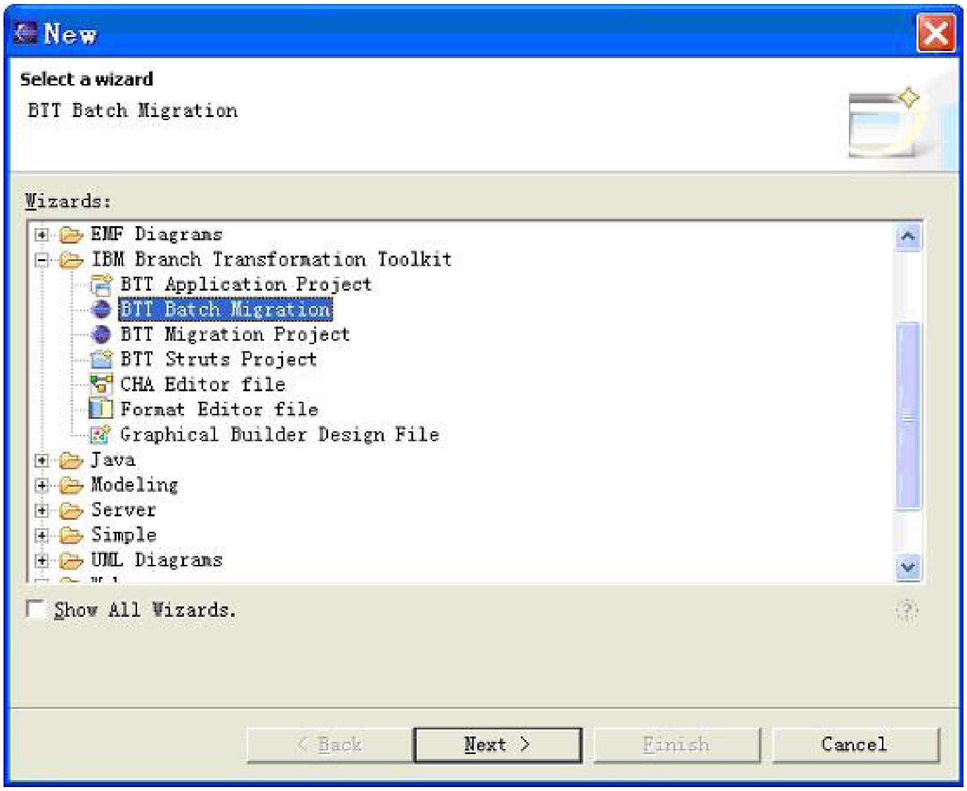
BTT provides you the batch migration function to help you migrate several of your applications at once.

To perform a batch migration, do the following:

1. From the WebSphere Integration Developer menu, select **File** → **New** → **Other**.



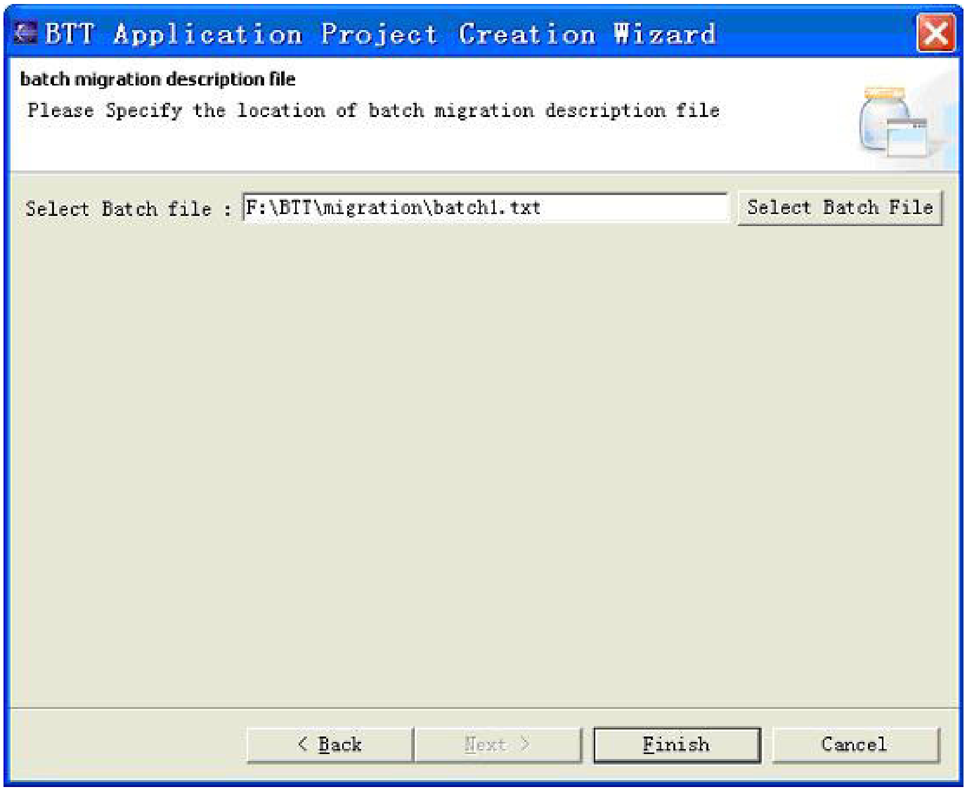
2. In the dialog box that pops up, expand **Branch Transformation Toolkit** → **BTT Batch Migration**, and then click **Next**.



3. In the BTT Application Project Creating Wizard, select the batch description file, which contains URLs of ear file and dse.ini file. Below is a sample of the batch description file:

   C:\BTT\migration\SampleApplication.ear

   C:\BTT\migration\DSE_SampleApplicationWeb\dse\server\dse.ini



batch migration description file

Please Specify the location of batch migration description file

Select Batch file : F:\BTT\migration\batch1.txt

Note: The migration tool can only migrate server-side application components. Make sure that the dse.ini file you selected is the server-side one.

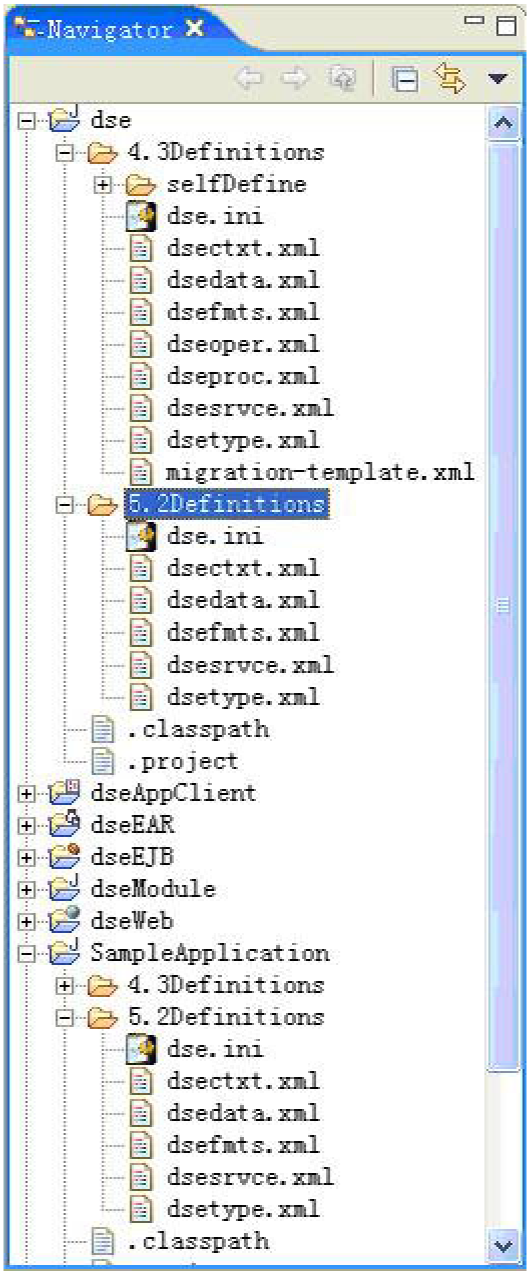
4. Click **Finish**.

After you create the migration projects in batch, you will find that all the BTT version 4.3 related definition files are copied and the projects of EAR, EJB, Web and Services are also created for every ear file and dse.ini file.

The wizard also migrates your version 4.3 dse.ini file into that of version 5.2, and copies the version 5.2 dse.ini file into the 5.2 definitions folder of your application migration project.

The wizard migrates the following files from version 4.3 to version 5.2:

- dsedata.xml
- dsetype.xml
- dsectxt.xml
- dsefmts.xml
- dsesrvce.xml



```
dse
  4.3Definitions
    selfDefine
    dse.ini
    dsectxt.xml
    dsedata.xml
    dsefmts.xml
    dseoper.xml
    dseproc.xml
    dsesrvce.xml
    dsetype.xml
    migration-template.xml
  5.2Definitions
    dse.ini
    dsectxt.xml
    dsedata.xml
    dsefmts.xml
    dsesrvce.xml
    dsetype.xml
  .classpath
  .project
dseAppClient
dseEAR
dseEJB
dseModule
dseWeb
SampleApplication
  4.3Definitions
  5.2Definitions
    dse.ini
    dsectxt.xml
    dsedata.xml
    dsefmts.xml
    dsesrvce.xml
    dsetype.xml
  .classpath
```

References

For the references related to migration from version 4.3 to version 5.2, see the following:

Manual works for migration

The migration tool does not migrate all of the things of version 4.3 to version 5.2 automatically. Users must do some things manually. They are:

- The base functionality of the migration tool will not migrate the logic inside the application code of the server operations and the actions of the flow process. If you do not customize the migration tool to include the process to move the logic to the generated code, you must manually move it.

- The services migration is not included in the migration tool. If there are any services that are out of the scope of version 5.2 in the version 4.3 application system, you have to migrate it manually.
- The server side event mechanism of version 5.2 is changed and is not migrated by the migration tool. You must migrate it manually.
- The communication service access is changed to use JCA connector. You must modify it in the Single Action EJB or the activity of Business process manually.

Related information

Events

Services in the application logic layer

Diagnosis for the migration tool

The migration tool will output the messages into a log file named BTT_Migration.log. The file is placed in the project workspace. When you get something wrong in the migration process, you can check the log file to see what happens and then do the necessary work to make the migration tool work.

Common Errors are as follows:

- The entity path is set as fromJAR. The migration tool does not support the path information set as fromJAR.
- Class not found. It is usually caused by not finding the jar file of the application in the plug-in directory, not setting the plug-in.xml file correctly, or the class not packaged in the jar files, or the project cache is not cleaned (when you start WID or RAD, you can attach ″-clean″ parameter to clean the cache).
- BTT envrionment initialization error. It is usually caused by not defining the defintion files correctly.
- xJSP path is not correct.

Notices

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

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

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

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

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION ″AS IS″ WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

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

Lab Director
IBM China Software Development Lab
Diamond Building, ZhongGuanCun Software Park, Dongbeiwang West Road No.8,
ShangDi, Haidian District, Beijing 100094 P. R. China

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

Trademarks and service marks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

AIX
AIX 5L
DB2
DB2 Universal Database
eServer
IBM
LANDP
MQSeries
pSeries
S/390 Parallel Enterprise Server
WebSphere
z/OS
zSeries

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

Other company, product or service names may be trademarks or service marks of others.