

IBM TRIRIGA Application Platform
3.7.0

Object Migration User Guide



Note

Before using this information and the product it supports, read the information in [“Notices” on page 35.](#)

This edition applies to version 3, release 7, modification 0 of IBM® TRIRIGA® Application Platform and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 2011, 2020.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

- Chapter 1. Migrating objects..... 1**
 - Object Migration overview 1
 - Object types.....1
 - Dependents2
 - Accessing the Object Migration tool..... 3
 - Object Migration page3

- Chapter 2. Exporting objects..... 5**
 - Object export overview..... 5
 - Object Migration export tips.....5
 - Creating an export package..... 6
 - Object Finder search parameters..... 7
 - Specifying objects to add to the export package..... 8
 - Adding objects to an export package.....8
 - Adding UX objects to an export package..... 9
 - Specifying a workflow to run during import.....9
 - Creating a file by using an export package.....10

- Chapter 3. Importing objects..... 11**
 - Uploading the migration package.....11
 - Preliminary validations.....11
 - Background uploads.....12
 - Background uploads with import.....12
 - Validating the migration package..... 13
 - Validation process..... 13
 - Package summary 14
 - Validating the package..... 14
 - Comparing objects.....17
 - Comparison methods.....17
 - Comparing all objects.....19
 - Comparing an object in an uploaded package to the target system.....19
 - Identifying objects that are identical.....20
 - Identifying existing objects.....20
 - Comparing two versions of the same workflow..... 20
 - Comparing two versions of the same UX content..... 21
 - Importing the migration package..... 21
 - Import Process 21
 - Object Migration import tips..... 22

- Chapter 4. Migrating database table indexes..... 27**
 - Exporting database table indexes..... 27
 - Exporting all indexes of a table..... 27
 - Exporting individual indexes..... 27

- Chapter 5. Configuring the Object Migration log..... 29**

- Chapter 6. Exporting workflows as text..... 31**

- Chapter 7. Downloading UX content..... 33**

Notices.....	35
Trademarks.....	36
Terms and conditions for product documentation.....	36
IBM Online Privacy Statement.....	37

Chapter 1. Migrating objects

When you develop or customize an application, you can migrate the objects such as business objects, forms, and workflows that represent your work to other IBM TRIRIGA Application Platform environments where they will be used.

Object Migration overview

You migrate objects from one TRIRIGA Application Platform environment to another by using a tool in the platform called Object Migration.

You first create an export package of selected objects and write them to a file. You then import that file into the target TRIRIGA Application Platform environment.

To migrate newer versions of the same objects that you previously migrated, you reuse the package that you previously created and repeat the steps.



Attention: Do not allow users to access the TRIRIGA Application Platform environment while you are processing Object Migration packages. User activity could result in data corruption or locked tables.

Object types

Some object types are supported by Object Migration while other object types are not.



Attention: IBM TRIRIGA does **not** support the customization of platform-managed objects, such as, but not limited to Document and Group objects. As a result, data that is added to any customized fields on platform-managed objects will **not** be supported by Object Migration import or export. For more information, see [Core objects in TRIRIGA Application Platform functionality](#).

The following object types are supported by Object Migration:

- Alternate Form List
- Application
- Associations
- Budget Token
- Business Object
- Calendar Set
- Document
- Form
- Group
- Hierarchy Structure
- List
- Module
- Navigation Collection
- Navigation Item
- Object Label
- Portal (post 3.0)
- Portal Section
- Query
- Record Data

- Scorecard (post 3.0)
- Styles
- Web Component
- Workflow

For more information about an object type, see the *Application Building for the IBM TRIRIGA Application Platform 3* collection of user guides.

You can select the Application or Web Component object type to search for and add applications or web components to your export package. By default, all metadata that is related to the application or web component, including dependent metadata, is automatically selected for inclusion in the package. You can use the search parameters to limit the dependent data.

You can use the Record Data option only for moving setup data, templates, and other items that are imported once and used as a template.

The following object types are not supported by Object Migration:

- Action item
- Manager
- Portal (pre 3.0)
- Scorecard (pre 3.0)
- Query list
- Query portal section
- Quick add list
- Shortcut list
- Site Style
- State Family

Note: Images that are stored in Record Data are not moved from the source to the target during the Object Migration export/import process. During export, Object Migration includes a simple path to the image file (in the source system's `userfiles` folder) in the Record Data XML. During import, this path is saved in the target system's record but the image is not visible to the user. For images to be available in the target environment after import, copy the image files from the source `userfiles` folder into the target `userfiles` folder.

Dependents

The concept of dependents is key to Object Migration. Object Migration copies objects from an environment or updates an environment with objects from a different environment. Any object can have one or more objects dependent on it.

For example, a business object can have several forms; in order for the forms to properly migrate, the business object must already exist in the target environment or be contained in the same package. This means that the form is a dependent of the business object. In other words, the form cannot be imported without the business object.

Note: Unless otherwise stated, selecting an object to include in the export does not automatically include any of the object's dependents.

Accessing the Object Migration tool

There are several methods for accessing the Object Migration tool.

About this task

- To access the Object Migration tool from the **Tools** menu, select **Tools > Administration > Object Migration**.
- To access the Object Migration tool from the Data Modeler, select **Tools > Builder Tools > Data Modeler**. Select **Object Migration** from the **Utilities** menu.

Object Migration page

An export package specifies a set of objects to be written out to a file so that it can be read by other TRIRIGA Application Platform environments. An import package contains objects from a file to be migrated into this platform environment.

The **Object Migration** page shows the status of each package, the date that each package was imported or exported, the user who created the package, and the package description.

To import an Object Migration package, click **New Import Package**.

To see the specifications for an existing Object Migration package, click the hyperlinked name.

To delete an Object Migration package, select the check box next to the package name and click **Delete**.

If you want to create a new export package that contains the same items as one of the listed packages, click the copy package icon next to the package name. On the **Create Package** page, change the name, and enter a description. **Create Mode** is preset to From Package. Only existing objects are put into the new package.

Tip: Use the copy package feature to reexport a set of objects that you previously imported. For example, use the copy package feature if you import a package into an environment, change the objects, and then migrate them somewhere else.

Note: Package names cannot contain special characters.

When you click **Create Package**, the system opens the **Object Migration** page.

Chapter 2. Exporting objects

You migrate groups of objects to other TRIRIGA Application Platform environments by first creating an export package and adding objects to the package. You then export the objects to a file and import the file into the target environments.

Object export overview

You create an export package and add objects and dependent objects to the package in the **Object Finder** panel of the Object Migration tool.

You view the objects that you add to your package in the **Navigation** panel of the Object Migration tool. Each object is added only once. Subsequent attempts to add the object are ignored.

You can use the Object Finder not only to add objects and dependents of those objects to an export package but also to view objects and dependents. You are not required to add any objects to a package to use the Object Finder.

Object Migration export tips

You can use these tips to better understand the Object Migration export process.

Object Migration area	Tip
Messages panel	Check the Messages panel for the status of the Object Migration process.
Navigation panel	When you click Objects in the Navigation panel, the Objects panel displays everything in the package. When you select one of the object types, the Objects panel displays everything in the package about that Object Type; for example, business objects might display triBuilding, triProperty, or triProposedSite.
Navigation panel hierarchy	To help the existing hierarchy load faster when you reopen an existing Object Migration package, all folders in the Navigation panel are initially closed. You do not need to expand the nodes to see the objects. When the hierarchy is loaded, click any folder name and view its objects in the Objects panel.
Object Migration page	The layout properties for the Object Migration panels are saved as cookies in your browser. If you sign on to another computer, you inherit the properties set up by the user of that computer. One way to reset the layout to the as-delivered default is to click Reset .
Record Data	If you are searching for Record Data, you must select a module. If you see a message, Nothing found , it is likely because a module is not selected. The Record Data display shows to which business object the Record Data belongs. Use the Record Data object type only for moving setup data, templates, and other items that are imported once and used as a template thereafter. Object Migration is not meant to move data from one environment to another.

Dictionary Record Data	<p>Use the Globalization Manager instead of Object Migration to export and import Dictionary (triDictionaryEntry) record data.</p> <p>If you choose to use Object Migration, make sure to remove existing Dictionary records from the target environment before you import the package that contains your Dictionary records. The Dictionary record name is mapped from a control number. The control number sequence in the source environment might not match the sequence in the target environment; therefore, duplicate records might occur on import.</p>
Group overrides	<p>In Navigation Builder, you can set a group override to hide navigation items in navigation collections from users in a security group that you designate. You can also set a navigation item to contain a specific target for a security group.</p> <p>Group overrides are exported as part of the navigation collections with which they are associated. In previous releases, group overrides are exported with their associated security groups.</p>
Security groups	Security group members are not included in the export process.
User Activity	Do not allow users to access the IBM TRIRIGA environment while you are processing Object Migration packages. User activity might result in data corruption or locked tables.
Workflows	<p>When you search for workflows, the results include all workflows regardless of their status. If a workflow is under revision, the last published or retired version is exported.</p> <p>When you add retired workflows to the Object Migration package and import them to another environment, the workflows are automatically retired in the target system.</p>

Creating an export package

You can create a package that does not initially contain any pre-loaded objects and then add only the objects that you want. You can also create a full package or a package that contains pre-loaded objects that were published on or after a specific date.

Procedure

1. Select **Tools > Administration > Object Migration**.
2. On the **Object Migration** page, click **New Export Package**.
3. On the Create Package form, enter a name and description for the new export package.
4. Select a value from the **Create Mode** drop-down box.

- To create a package that initially does not contain any pre-loaded objects, select **Empty**.
- To create a package that contains all objects, select **Full Metadata Package**.

The package will contain all objects from the entire system, except for most record data. The record data for Classifications is included in the package; however, any record data that is associated to the classifications is not included.

- To create a package that contains all objects that were published on or after a specific date, select **By Date**. Select a date in the **Modified After (Date)** field and select the **First Level Dependencies** check box.

Record Data is included only if it is a dependent of another object in the package. The system uses a recursive find to capture a single level of dependents.

5. Click **Create Package**.

What to do next

If you selected **Empty**, the package does not contain any objects and the next step is use the Object Finder to locate and specify the objects to include.

If you selected **Full Metadata Package**, the package contains all necessary objects in the environment and is ready for export.

If you selected **By Date**, the package contains pre-loaded objects, and the next step is to add or delete objects.

Object Finder search parameters

You can select any one search parameter or a combination of search parameters to filter your search to specific groups of objects.

As you gain familiarity with your object data in the Object Finder, you can speed up your use of the search parameters. Depending on the parameter that you select, the options that are available for the other unselected parameters are updated. Instead of choosing an object type first, for example, you can choose one of the other parameters first.

Search parameter	Description
Object Type	Filter the search to objects of a specific object type.
Module	Filter the search to objects in a specific module. This parameter does not display for the object types Application, Budget Token, Document, Form Style, Group, Module, Navigation Collection, Navigation Item, Portal, Portal Section, and Web Component.
Business Object	Filter the search to a specific business object. This parameter displays if you select a module that contains business objects.
Modified After (Date)	Filter the search to objects modified after a specific date. If you do not select a date, objects are returned regardless of when they were modified.
Modified By (User)	Filter the search to objects that were modified by a specific user. If you do not specify a user name, objects are returned regardless of the user who last modified them. Some object types do not support this parameter in their database layout. For those object types, the system does not return any objects if you identify a user for this parameter. The supported object types for this parameter are Application, Business Object, Document, Form, Group, Module, Navigation Collection, Navigation Item, Query, Record Data, Web Component, and Workflow.

Table 2. Parameters in the Search Parameters section of the **Object Finder** panel (continued)

Object Label	<p>Filter the search to objects that contain a specific object label.</p> <p>This parameter displays for the object types All, Application, Business Object, Form, Module, Report, Web Component, and Workflow.</p> <p>When All is selected, only Application, Business Object, Form, Module, Report, Web Component, and Workflow objects are searched, as they contain object labels. The objects that are not labeled are not searched.</p> <p>This parameter searches for only the objects that currently contain the object label. In Object Label Manager, the Labeled Objects tab for the object label definition contains not only objects that currently have the label but also objects that previously had the label.</p>
Object Name	<p>Filter the search by entering an object name that you want and selecting a filter.</p>

Specifying objects to add to the export package

The process of adding objects to the export package is the most complicated part of the migration preparation. It involves including the right objects in the package. It also might involve excluding objects from the package.

Adding objects to an export package

The next step is to add objects to your export package by using any of the available search parameters in the **Object Finder** panel.

About this task

You can add one object at a time to the package and then add its dependents, or you can add a set of objects and then add their dependents. The method that you choose to add objects that are dependents of other objects to your package is based on your objectives and your familiarity with your IBM TRIRIGA implementation.

It might not be necessary to add all dependencies. It entirely depends on your target environment. You might need to add only the objects that changed since a specific date. Using the Modified After or Modified By search parameters can help.

When you locate the first level of dependencies for the Record Data object type, all subsequent levels of associated record data are retrieved in the dependency results for the Record Data that you selected. Therefore, there is no need to find dependents of dependents for Record Data.

Procedure

1. Locate an object or a set of objects to add to the export package by specifying one or more parameters in the **Search Parameters** section of the **Object Finder** panel.
2. Click **Search**.
3. To add the object or set of objects that are listed in the Search Results section to the export package, select the check boxes for the objects and click **Add Selected Objects**.
4. To add dependents of the selected object or objects that you added to the package, complete the following steps.
 - a) Locate dependents of the selected object or objects by clicking **Find Dependents**.
 - b) Add dependent objects that are listed in the Search Results section to the export package by selecting the check boxes for the dependents objects and click **Add Selected Objects**.
5. Add dependents of the dependent object or objects that you added to the package.
 - a) Locate dependents of the selected dependent object or objects by clicking **Find Dependents**.

- b) Add dependent objects that are listed in the Search Results section to the export package by selecting the check boxes for the dependent objects and clicking **Add Selected Objects**.
6. Repeat the process of adding dependents of dependents as needed.
7. Click **Reset Search** in the Object Finder to clear the current search data and select the search parameters for your next object or set of objects.
8. When the export package contains all of the objects that you want, save the package.

Adding UX objects to an export package

You can select the **Application** or **Web Component** object type to search for and add UX applications or web components to your export package.

About this task

By default, when you select **Search**, all metadata that is related to the application or web component, including dependent metadata, is automatically selected for inclusion in the package.

An **Include Dependents** check box is displayed when the search completes and the search results contain applications or web components. The check box is selected by default. If you want to limit the dependent data such as Queries, Business Objects, and Workflows in the package, deselect the **Include Dependents** check box. Then, use the search parameters to search for and add the dependent data that you want. UX dependents of applications and web components, such as Models and Data Sources, are always included at the time you export the package.

The **Include Dependents** check box is displayed only if your search results contain application or web component object types. It is only with these object types that dependent data can be automatically selected for inclusion in a package. If your search contains other objects that are not associated with an application or web component, the **Include Dependents** check box has no bearing on those objects. For those objects, you must use the standard method of using **Find Dependencies** to locate and add the dependent data that you want.

If you select the **Application** or **Web Component** object type and add a search term to the **Object Name** field and select **Search**, the term is cleared after the results are displayed. The term is cleared so that all dependent objects that are associated with the application or web component are selected, not only those objects whose name contains the term. If you want to go back to your search criteria, select **Back**.

To track your latest changes before you perform the export, you can create revisions for any UX metadata objects that you changed and saved, such as an **Application**, **Model**, or **Web View**. The **Create Revision** action saves your changes and applies the **In Progress** object label to the UX metadata component and all of its dependent children components. Then you can apply a custom object label to these modified objects in the Object Label Manager. If needed, as you add objects to the export package, create revisions by drilling down into the highlighted objects and clicking **Create Revision**.

Specifying a workflow to run during import

When you add workflows to your export package, you can select a workflow that automatically runs on import. When the import of the package in your target TRIRIGA Application Platform environment completes successfully, the workflow runs.

About this task

The **Property** panel on the Object Migration page contains a Workflow drop-down, which gets populated when you add workflows to your export package in the **Object Finder** panel. If you do not add workflows to the package, the Workflow drop-down is blank.

Procedure

1. After you add one or more workflows to your export package in the **Object Finder** panel of the Object Migration tool, click the **Workflows** drop-down in the **Property** panel.

2. Select the workflow that you want to run on import.
3. Save the package.

Creating a file by using an export package

When an export package contains the objects that you want, you create a .zip file that you will use to migrate your changes from one environment to another.

Procedure

1. With the export package open in the Object Migration tool, click **Export**.
2. In the **Objects** panel, select **Wait** for a synchronous export or **Background** for an asynchronous export. TRIRIGA Application Platform generates a .zip file that contains a .xml file for each of the package's objects.
3. If you are exporting synchronously, specify the location for the exported file and click **Save**.
4. If you are exporting asynchronously, the file is saved in the userfiles\ObjectMigration directory of the TRIRIGA Application Platform installation, for example, C:\Tririga\userfiles\ObjectMigration\Building B0.zip.

While the Object Migration Agent is working on the export package, you cannot change the definition of the package shown in the Object Migration screen. You are notified when the export is completed.

What to do next

During the export process, TRIRIGA Application Platform might log error and warning messages in the Object Migration log file. The Object Migration log file is in the log directory of the TRIRIGA Application Platform installation, for example, C:\Tririga\log\ObjectMigration.log.

When the .zip file is downloaded, the status of the Object Migration package changes to Exported. You are ready to migrate the objects into another TRIRIGA Application Platform environment by importing the .zip file into that environment.

Chapter 3. Importing objects

After you export objects to a .zip file, you can import the file into other IBM TRIRIGA Application Platform environments.

Before you begin

Do not do normal production activities in the IBM TRIRIGA Application Platform environment while you are importing an Object Migration package into that environment. Import the package when the environment has minimal system activity.

Before you start the import steps, ensure that the Cleanup Agent is not running. When you finish the import steps, allow the Cleanup Agent to run as it was running previously. Make sure that production mode has the proper value and restart the server.

If you have questions about these concepts or procedures, see the *IBM TRIRIGA Application Platform 3 Installation and Implementation Guide* and *IBM TRIRIGA Application Platform 3 Administrator Console User Guide*.

Uploading the migration package

The first step for importing the package into the target TRIRIGA Application Platform environment is to upload the .zip file that you created during the export process.

About this task


To upload the package, you select **New Import Package** on the **Object Migration** page. The uploaded package is added to the Import Packages section of the page with a status of **New**.

When you open the package, the objects in the .zip are extracted and preliminary validations are completed on the objects. The objects and validation information are displayed in the **Navigation** panel on the **Object Migration** page. Any temporary files that are created during the package upload process are deleted automatically.



Preliminary validations


When you upload the package into the target TRIRIGA Application Platform environment, preliminary validations are completed on the objects.

The first preliminary validation that is done during the upload process checks the following items.

- If a module is specified in an uploaded object, whether that module exists in the package or in the system.
- If a business object is specified in an uploaded object, whether that business object exists in the package or in the system.
- If a dependent item that is specified in an uploaded object is not in the package or the system, the object is tagged as Invalid and the object name is red. An  icon is displayed next to the object name.

The second preliminary validation checks if the object name, including the module and business object if appropriate for that object type, is in the target system.


- If the object name is in the target system, and their object labels do not cause a conflict, an  icon is displayed next to it and the object name is in blue text.
- If the object name is not in the target system, an  icon is displayed next to it and the object name is in green text.

If an object name has an  icon and is in green text, but it exists in the target system, check the exact spelling of the object name. If an object name is blue but it does not exist, check the exact spelling of the object name.

If the object type has a module and business object, check the module name and business object name for extra spaces or a difference in capitalization.

If the target system contains more than one object with the same name, a warning is written to the Object Migration log. For example, a query at the module level and a query in one of the module's business objects might both exist. The import process updates the object in the target system that has the lowest ID with the object from the import package.

As an extension of the second preliminary validation, if the object name is in the target system, the process checks if the object label for an object in the package conflicts with the object label for that object in the target system.

- If the object label in the package and the target system are both IBM TRIRIGA labels, both custom labels, or both Root labels, then there is no conflict.
- If the object label in the package is an IBM TRIRIGA label and the object label in the target system is a Root label, then there is no conflict.
- If there is no object label in the package, then the **Validate** action is necessary. If validation reveals a label in the package, then use the same comparison for the above cases.
- Otherwise, if the above cases do not apply, then there is an object label conflict. An  icon is displayed next to the object and the object name is in orange text.

For more information about object labels, see [Object Labels and Revisions](#) in the [IBM TRIRIGA wiki](#).

Background uploads

You can choose to upload Object Migration packages as a background process.

You copy the file to be uploaded into the `userfiles\ObjectMigration\Uploads` folder of the installation folder. An example of this location in Microsoft Windows is `C:\Tririga\userfiles\ObjectMigration\Uploads`. When you add a file to this folder, the Object Migration Agent detects the file and automatically creates a Package for the file.

After the Object Migration package is created, the system automatically deletes the file.

Background uploads with import

You can choose to upload and import Object Migration package zip files as a background process.

To start the background process, you copy the import package zip file into the `userfiles\ObjectMigration\UploadsWithImport` sub folder of the installation folder. The Object Migration Agent detects the file, then automatically uploads and imports it.

After the Object Migration package zip file is imported, the file is automatically deleted from the `\UploadsWithImport` folder.





An example of the installation folder location in Microsoft Windows is `C:\Tririga\userfiles\ObjectMigration\UploadsWithImport`.

Validating the migration package


The purpose of the validation step is to detect problems caused by differences between the uploaded objects and objects in the target platform environment.


About this task

To accurately assess the objects, you must know about both the source and the target systems. The information is based on a comparison of the XML in the package with the target system. You must also know the meaning of the object text colors and icons.

Object color and icon	Description
Object in green text with the  icon.	Object is in the package, but there is no object with the same object name and module and business object names, if appropriate, in the target environment.
Object in blue text with the  icon.	Object with the same object name and module and business object names, if appropriate for the object type, is in both the package and the target environment, and their object labels do not cause a conflict.
Object in red text with the  icon.	Object dependent items are neither in the package nor the target environment. The issue must be resolved before the package can be imported.
Object in orange text with the  icon.	Object with the same object name and module and business object names, if appropriate for the object type, is in both the package and the target environment, but their object labels cause a conflict.


Procedure

1. Click the  icon in the **Navigation** panel to show the details of the uploaded package.

Note: A content object's name is always green with a  icon (indicating new) in the Object Migration screen.

Note: If the target system contains more than one object with the same name (for example, a query at the module level and a query in one of the module's business objects), a warning is written to the Object Migration log. The import process updates the object in the target system that has the lowest ID with the object from the import package.


2. To see an explanation for an object, click the hyperlinked name of the object in the **Navigation** panel. The **Objects** panel shows a brief Description.

Note: If red objects with a  icon exist, you cannot import the migration. You must resolve each red problem before you can import the object into the target platform environment.

Validation process

During the validation process, the system analyzes the uploaded XML from the .zip, and checks for invalid or missing references to other objects.

For example, if a business object has a pre-create workflow, the validation process checks to make sure that workflow is in the package or the system. If not, the system logs a warning against that business object.

When the analysis discovers more serious situations, the system logs an error against the object and renders it Invalid (shown in red with an  icon). One example is when a business object type is changed from standalone to embedded. The TRIRIGA Application Platform system does not allow this change; therefore, Object Migration considers it an error. You must resolve all Invalid objects before you can import the package.

Note: The system always validates all objects before it reports a successful or failed validation. This process allows the person responsible for importing the package to see all of the issues (warnings and errors) at hand.

Package summary

To see all warnings and errors that are found during validation, click the **Summary** action on the top of the Object Migration screen.

The Object Summary at the beginning of the Package Summary report lists statistics for each object type. The Warnings and Errors are hyperlinked to the corresponding summary section further in the report.

The body of the Package Summary report contains a separate section for each object type with warnings and errors. Within the sections, you see the object name, module, and business object if appropriate for the object type, and the corresponding warning or error messages.

Table 4. Package Summary example


Name:	TRIRIGA_10_11172010_After_0323							
Description:	TRIRIGA_10_11172010_After_0323							
Status:	Imported							
Object Summary								
	Type	# of Objects	New	Existing	Valid	Invalid	Warnings	Errors
1	Budget Token	30	-	30	30	-	-	-
2	Business Object	354	-	354	354	-	-	-
3	Content	82	82	-	82	-	-	-
4	Document	268	-	268	268	-	2	-
5	Form	430	-	430	430	-	-	-
6	Group	59	-	59	59	-	1	-
7	Hierarchy Structure	13	-	13	13	-	-	-
8	List	290	-	290	290	-	-	-
9	Module	53	-	53	53	-	-	-
10	Navigation Collection	171	-	171	171	-	-	-
11	Navigation Item	1441	-	1441	1441	-	-	-
12	Portal	89	-	89	89	-	-	-
13	Portal Section	367	-	367	367	-	-	-
14	Query	2583	-	2583	2583	1	-	1
15	Record Data	780	-	780	780	-	-	-
16	Scorecard	23	-	23	23	-	-	-
17	Styles	13	-	13	13	-	-	-
18	Workflow	1063	-	1063	1063	-	-	-
		8109	82	8027	8109	1	3	1
Document Warnings								
	Object	Module	Business Object	Message				
1	\\ROOT\\TRIRIGA\\Item\\Critical To Quality (CTQ) Item Response Record	Document	Document	1) Error updating Existing Document [\\ROOT\\TRIRIGA\\Item\\Critical To Quality (CTQ) Item Response Record].				
2	\\ROOT\\TRIRIGA\\Location\\Floor_Space_Rollup	Document	Document	1) Error updating Existing Document [\\ROOT\\TRIRIGA\\Location\\Floor_Space_Rollup].				
Group Warnings								
	Object	Module	Business Object	Message				
1	\\Admin Group	Group	Group	1) Admin Group will not be imported.				
Query Errors								
	Object	Module	Business Object	Message				
1	triTaskHoldLog - Display - triTask	triLogging	triLogging	1) Could not find Module [triLogging] in System or Package.				

Validating the package


When the import package contains only the objects you want to add to the system, you validate the package. In some cases, such as in packages created prior to IBM TRIRIGA Application Platform 3.5.4, if

there is no object label in the package, then the **Validate** action is necessary to determine the label and check for an object label conflict.

Procedure

1. Click **Validate**.
2. In the **Objects** panel, the system asks if you would like to process the validation synchronously (**Wait**) or asynchronously (**Background**).
 - If you select **Wait**, the status in the **Property** panel changes when the validation completes.
 - If you select **Background**, a notification is sent when the validation process is complete.
 - The **Background** notification shows in the Notification section on the Home Portal.
 - When the validation is finished, some of the objects in the **Navigation** panel might be shown in red. Note the  icon. This notation means that there is a problem with the object. To see an explanation of the problem, click the name of the object.
 - A brief description of the problem shows in the **Objects** panel.

Results

If any objects contains a red  icon after validation, the objects are Invalid and you cannot import the migration. Each invalid object must be resolved before you can import the objects into the target platform environment.

Troubleshooting invalid objects

A migration object has an invalid reference when it refers to objects that do not exist in TRIRIGA Application Platform. You can resolve invalid object issues by using two methods.

Depending on the criticality, some invalid references are detected as errors, and others are shown as warnings. When such non-critical situations occur, the platform still imports the migration objects. However, it also logs warning messages to the Object Migration log file to indicate a missing object and where it is referenced.

Warnings do not stop the import from the proceeding and are considered less critical than errors. Only errors or invalid objects cause the import to fail. However, do not ignore warning messages. Warnings can lead to problems with your applications after you import. Make sure that you understand the cause of the warning before you continue.

The warning message format consists of a name identifier of the object that has the invalid reference, a name identifier of the missing object, and how it is referenced. In order to help you quickly determine which object is having an issue, the platform uses all available information about an object to construct its name identifier. Usually an object's name identifier is combined with its module name, its business object name (if any), and its name. For example, `triCostItem_triPaymentLineItem_triPaymentOneTime` form name identifier consists of the module name (`triCostItem`), business object name (`triPaymentLineItem`), and the form name (`triPaymentOneTime`).

Method	Description
Delete the problem object.	<p>You can delete an object when it is displayed in the Objects panel by checking its check box and clicking the Delete action in the Objects panel. Blindly removing objects from the package can cause further problems during validation and import.</p> <p>For example, if you remove a new business object or module from the package and other objects in the package reference this business object, the next time you validate, these other objects become invalid.</p>

Table 5. Methods of resolving invalid objects (continued)

Fix the underlying problem and then validate the objects again.	For example, if the problem is a missing business object, re-export the package that contains the missing business object. Or manually create the required business object in the target platform environment and run the validation again.
---	---

The following table is a partial list of the errors that can be flagged during validation.

Table 6. Partial list of validation errors and their descriptions

Object type	Brief error description	Error detail
Business Object	List does not exist.	The business object in the migration objects has a list field that refers to a list of values that does not exist in the migration objects or the target platform environment.
Business Object	Locator BO does not exist.	The business object in the migration objects has a locator field that references a business object that does not exist in the migration objects or in the target platform environment.
Business Object	Published name is missing.	The business object does not have name mapping information specified.
Business Object	[Object name] is invalid. Existing section data not valid.	This business object exists in both the migration objects and the target platform environment. Both versions of the business object have a record section that is named xxx, but the properties of the two versions of the section are different. The possible differences might include: <ul style="list-style-type: none"> • The 2 versions of the record section reference a different business object. • One of the versions of the record section is a single-record section and the other is a multiple-record section.
Business Object	xxx is invalid. Reference BO does not exist.	This business object has a record section named xxx. The business object that is referenced by the xxx section does not exist in the migration objects or the target platform environment.
Form	Invalid BO.	This form in the migration object is based on a business object that does not exist in the migration objects or the target platform environment.
Workflow	Attach format file task's source is missing.	This workflow in the migration objects contains an Attach Format File task that refers to a Document record that does not exist in the migration objects or the target platform environment.
Workflow	Call Validation Workflow on LI task's Source is missing.	This workflow in the migration objects contains a Call Workflow task that refers to another workflow that does not exist in the migration objects or the target platform environment.
Workflow	Query task's Source is missing.	The workflow in the migration objects contains a Query task that refers to a query that does not exist in the migration objects or the target platform environment.

If the target system contains more than one object with the same name (for example, a query at the module level and a query in one of the module's business objects), a warning is written to the Object Migration log. The import process updates the object in the target system that has the lowest Id with the object from the import package.

The following text shows messages that are typically logged by the validation step.

```
Validation of the Test package begins

Business object validation
triProject_triCapitalProject business object is referencing to invalid object(s):
1) triNotificationAction_triInsuranceContractNotification as an associated
business object.

Form validation
triCostItem_triPaymentLineItem_triPaymentOneTime form is referencing to
invalid object(s):
1) triContract_Blank business object name_triRealEstateContract - Find -
List of Real Estate Lease and Owned Property as Find action popup query
of triRealEstateLeaseContract section of triGeneral tab.

Query validation
triContract_triRealEstateContract_triRealEstateContract - CRYSTAL -
Future Rent Obligations Detail query isvreferencing to invalid object(s):
1) \ROOT\TRIRIGA\Contract\Future Rent Obligation Detail_Document_Document
as an invalid crystal report template.

List validation

Manager validation
The validation is complete
```

According to this sample Object Migration log, the validation step detected the following invalid references:

- The `triProject_triCapitalProject` business object has an association with the `triNotificationAction_triInsuranceContractNotification` business object, but such business object does not exist in the platform. As a result the `triProject_triCapitalProject` business object is created without the association.
- The `triCostItem_triPaymentLineItem_triPaymentOneTime` form references the `triRealEstateContract - Find - List of Real Estate Lease and Owned Property` query as its section popup query action. However, the query does not exist in the platform. The message also specifies the name of the referencing section, in this case, the `triRealEstateLeaseContract` section of the `triGeneral` tab.
- The `triContract_triRealEstateContract_triRealEstateContract - CRYSTAL - Future Rent Obligations Detail` Crystal Report query references an invalid Crystal Report template.

Comparing objects

The next step of the import process is to compare the objects that you uploaded with the objects in the target TRIRIGA Application Platform environment.

Comparison methods

Object Migration offers two comparison methods to find out more information about the differences between uploaded items and existing items in the system.

You can compare the entire upload with the **Comparison Report** action or compare individual items with one of the **Compare** actions in the **Objects** panel. The comparison logic is the same for the comparison report and an individual compare. The main difference is that the comparison report compiles the result into a file and the individual compare method displays the result in the user interface.

The following object types are compared:

- Alternate Form List
- Budget Token

- Business Object
- Calendar Set
- Form
- Form Style
- Hierarchy Structure
- List
- Module
- Navigation Collection
- Navigation Item
- Portal
- Portal Section
- Query

The following object type is partially compared:

- Workflows.

Object Migration does a partial comparison for Workflows. The Workflow header is compared and details on the differences between the source and target, if any, are reported. Workflow tasks are also compared; however, the report indicates only if the tasks are the same or different between the source and target. Details on differences are not reported. When the report indicates "No Differences" on Workflows, this means that all header properties, as well as the Workflow tasks, are the same between the source and target.

For a detailed examination of Workflows, use the **Text Export Selected** action in the Objects panel of the Object Migration tool.

The following UX object types are compared:

- Application
- Action
- Action Group
- Data Source
- Data Source Field
- Model
- Model And View
- Web Component
- Web View File

Files that are attached to the Web View File metadata, such as HTML and CSS files, are included in the compare. However, the results indicate only if the files in the source and target are the same or different. Details on differences are not reported. A report that indicates No Differences on Web View Files means that all properties, as well as any attached files, are the same. You can use a HTML "diff" tool to compare file versions.

The following object types are not compared:

- Content. The information is binary.
- Document. The information is binary.
- Group. Some groups are enormous and a comparison might take several hours to run.
- Record Data. The definition is a dynamically structured XML-based on sections, fields, and other elements.

When Content, Document, Group, or Record Data object types are compared, the result is always **No Differences**. Do not be falsely assured by a **No Differences** comparison result for existing Content, Document, Group, and Record Data object types.

Comparing all objects

You can create a comparison report that compares every object in the uploaded package with objects in the target system. The result is a tab-delimited .txt file. You can save comparison reports for future analysis and comparison.

About this task

Each row in the comparison report represents one difference or conflict between the uploaded package and the target system. For example, if a business object has 10 differences, 10 lines display in the comparison report.

Procedure

1. Click **Comparison Report** on the **Object Migration** page.
2. In the **Objects** panel, create the report.
 - Select **Wait** to create the report synchronously. Specify a location to download the .txt file and select **Save**.
 - Select **Background** to create the report asynchronously. The .txt file is saved in the userfiles\ObjectMigration directory of the IBM TRIRIGA installation; for example, C:\Tririga\userfiles\ObjectMigration\triLog Query Import.txt. You might need assistance from your IT department to gain access to the file because it is on the server.
3. Review the comparison report.
You can import the report into Microsoft Excel and analyze the information; for example, you can convert the header row into filters and use filtering for areas of interest.





Comparing an object in an uploaded package to the target system

The **Compare** action looks at only one object and compares what is in the uploaded package to what is in the target system.

About this task


To accurately assess the objects, you must know about both the source and the target systems. The information is based on a comparison of the XML in the package with the target system. You must also know the meaning of the object text colors and icons.

Table 7. Description of object text colors and icons

Object color and icon	Description
Object in green text with the  icon.	Object property exists in the package, but not in the target environment.
Object in blue text with the  icon.	Object property exists in the target environment, but not in the package.
Object in red text with the  icon.	Object property exists in both the package and the target environment, but are different.
Object in orange text with the  icon.	Object with the same object name and module and business object names, if appropriate for the object type, is in both the package and the target environment, but their object labels cause a conflict.

To use the **Compare** action, do the following steps.

Procedure

1. To compare one of the objects in the upload with its counterpart in the system, select the object in the **Navigation** panel. The object displays in the **Objects** panel.
2. Select an item in the **Objects** panel and click **Compare**.
The result is displayed in the Description column of the **Objects** panel.
3. Click the  icon to expand and close the object levels.
 - An object in red text indicates that the object exists in the target but not the package.
 - An object in green text indicates that the object exists in the package but not the target.

Identifying objects that are identical

The **Compare & Select Matching Objects** action identifies objects that the Object Migration comparison method considers identical between the source package and the target environment. This action provides a quick way to remove these objects, leaving only the objects that changed.

About this task

To compare one of the objects in the upload with its counterpart in the system to identify matching objects, do the following steps.

Procedure

1. Select the object in the **Navigation** panel. The object displays in the **Objects** panel.
2. Select one or more items in the **Objects** panel and click **Compare & Select Matching Objects**.
3. Review the result in the Description column of the **Objects** panel.

Identifying existing objects

The **Select Existing Objects** action identifies objects that the Object Migration comparison method finds in both the source Object Migration package and the target environment.

Procedure

1. To see whether one of the objects in the upload exists in the system, select the object in the **Navigation** panel. The object displays in the **Objects** panel.
2. Select one or more items in the **Objects** panel and click **Select Existing Objects**.
3. Review the result in the Description column of the **Objects** panel.

Comparing two versions of the same workflow

The **Text Export Selected** action exports one or more workflows as text. You can export two versions of the same workflow as text and compare them to decide whether you want to import a workflow from a migration package.

Procedure

1. Select **Workflow** in the **Navigation** panel.
2. Select one or more workflows in the **Workflow** panel and click **Text Export Selected**.
3. Save the file to the system.

Comparing two versions of the same UX content

The **Download Content For Selected** action downloads one or more UX content files, such as .html and .css files. You can download two versions of the same UX content and compare them to decide whether you want to import a content file from a migration package.

Procedure

1. Select **Web Component** or **Web View File** in the **Navigation** panel.
2. Select one or more UX components or content files, respectively, in the **Objects** panel and click **Download Content For Selected**.
3. Save the files to the system.

Importing the migration package

After you validate a package and there are no invalid (red) objects, you are ready to import. The import might take some time to process.

About this task

The Object Migration Agent must be running for the import processing to occur.

When you click **Import**, the platform adds a message in the **Objects** panel that indicates that the import is started.

Note: You can also click **Import Non-Conflicting** to import new objects and objects with non-conflicting object labels only. If an object has a conflicting object label, or an object is not new but is not supported by object labels, then the object will not be imported. Be aware that this process can potentially put your environment in an incomplete upgraded state as necessary objects may not be imported.

You receive a notification in your portal Home page when the import is complete. The status also displays in the **Import Packages** section of the Object Migration screen.

If the package is configured to automatically run a particular workflow, it runs when the import process completes. You optionally specify a workflow to run at import completion when you create the package in the source IBM TRIRIGA Application Platform environment.

In the import package, you can deselect the workflow that was set in the export, but you cannot select other workflows in the package. When the workflow runs upon the completion of the import, a record is created based on several rules.

- If the workflow is defined on a business object, then a record of that business object is created.
- If the workflow is not defined on a business object, then a record is created of the base business object of the module.
- If the workflow is not defined on a business object and the module does not have a base business object, the run fails and therefore, the import fails.

The empty record is created for workflow debug purposes. If the workflow instance is turned on, you can view the workflow instances through the empty record.

Import Process

Before the import starts, the system revalidates all of the objects in the package. The revalidation allows the system to make sure that it is always using the most current snapshot of what is contained within the system.

The revalidation is important if a package is uploaded several days before it is imported. During this lag time, anything can happen. For example, objects that are identified as new within a package could be manually created or imported as part of a different package. This could cause much confusion if the Import process tries to recreate the same object. The revalidation is one of the most important stages of the entire process.

Note: The import validation is the same process that can be triggered manually by the user.

After the revalidation, if any errors are encountered, the import stops and the package status changes to Import Failed.

If no errors occur, the process imports the objects in the following order:

- Content
- Document
- Form Style
- Budget Token
- Record Data for UOM_Type and UOM_Values
- Module
- Business Object
- Form
- Record Data (excluding UOM_Type and UOM_Values)
- UX components (Application, Web Component, and related UX metadata)
- Hierarchy Structure
- Query
- Workflow
- Navigation Collection
- Navigation Item
- Group
- Portal
- Portal Section
- List

To help resolve any circular dependencies/references, the system automatically imports the package two times. After each pass, the system refreshes all system caches on the environment in which the import is being performed. If there are other running process servers or application servers, the caches on those systems should be manually refreshed.

During import, if an Application or Web Component object exists in the target environment, a revision of the object is created in the target environment. To prepare a clean target, all UX metadata that is related to the specific Application or Web Component object in the target environment, but is not included in the import package, is deleted.

Object Migration import tips

You can use these tips to better understand the Object Migration import process.

Object Migration area	Tip
Associations	The Object Migration process merges associations. Any new associations stay intact.
Business Object Type (general information)	Business object type information is overwritten by the import process.
Cache	In past implementations of Object Migration it was necessary to clear / refresh the system's cache after a successful import. It is no longer necessary to clear the cache. The import process automatically refreshes the cache.

<i>Table 8. Object migration import tips (continued)</i>	
Content objects	These objects are used and referenced by Record Data and Form objects and are added automatically as required dependents during the Export process. During import, all content objects show as new objects (green). If a content object in the source has the same name as a content object in the target, a new content object is created in the target during import.
Default form	When the form is imported, the default flag is set to the form that is imported. You might want to reset it.
Duplicate objects	If the target system contains more than one object with the same name, a warning is written to the Object Migration log. For example, say a query at the Module level and a query in one of the module's business objects. The import process updates the object in the target system that has the lowest ID with the object from the import package.
Dynamic list	All dynamic list information is overwritten by the import process.
Field mapping	Field mapping is overwritten, with 1 exception. If a control start number of the imported field mapping is smaller than a current control number, the current control number is retained.
Fields	<p>The Object Migration process merges business object fields. All modifications to existing fields in the business object are overwritten with several exceptions:</p> <ul style="list-style-type: none"> • For text fields, if the size of the imported field is smaller than the current size, the current size is not reduced. • For text fields, if the size of the imported field is 0, the current size is not reduced. If the imported field is new, a default size of 700 is used. <p>Whenever possible, change existing fields in the forms. If the required and the read-only attributes are set in the Data Modeler, they cannot be overridden in the form.</p> <p>If changes are made to existing field properties prior to the import, these changes need to be redone after the import.</p>
Forms	All form information is overwritten by the Object Migration import. To preserve any modified form, rename the form. To retain the original form, make a copy of the form before you change it and rename it back to the original name.
Import twice	In past implementations of Object Migration it was necessary to perform every import twice. The import process now automatically imports the Object Migration package twice.
Language tables	When an object migration package that was created before TRIRIGA Application Platform 3.3 is imported into an environment that was upgraded to TRIRIGA Application Platform or later and IBM TRIRIGA 10.3 or later, the language table is deleted.
Navigation collection / Navigation item	<p>All navigation collection/navigation item information is overwritten by the import process. To preserve any of your modifications to the managers:</p> <ul style="list-style-type: none"> • Create a navigation collection/navigation item. • Remove the navigation collection with the <code>tri</code> prefix from the Navigation Builder. • Add the new renamed navigation collection item to the Navigation Builder. <p>Alternatively, if the changes are minimal, you could make note of the changes and reapply after the import. In some circumstances, this alternative method might be the quicker option.</p>
Module	Existing modules are not changed by the import process and any new modules are added.
Portal	When an imported portal has the same name as an existing portal, all portal information is overwritten by the import process.

<i>Table 8. Object migration import tips (continued)</i>	
Pre-Create workflow	If a pre-create workflow is modified, after an import you need to repoint this attribute in the Data Modeler to the renamed workflow. The import process resets the attribute to a workflow with the prefix <code>tri</code> .
Publish name	If the publish name is modified prior to the import, redo the modification after the import.
Queries/Reports	All query/report information is overwritten by the import process. To preserve any modified queries/reports, rename the query/report. To retain the original query/report, make a copy of the query/report before you change it and rename it back to the original name.
Record Data	<p>All record data information is overwritten by the import process.</p> <p>With older versions of Object Migration, the Record Data XML was not exported with the name of the form or the object state, which causes unpredictable behavior during import.</p> <p>If new Record Data does not have a Form name, the system uses the default form for the business object. Also, if the object state is not specified, the system chooses an arbitrary action to transition the new Record Data from the null state.</p> <p>The migration package includes information for all associations that are discovered in the source environment for Record Data included in the package. Associations for Record Data included in the package are established for the target environment during the migration package import process.</p> <p>When you create the package, you can choose to not include association information for any record. Open the panel for the Record Data included in the package. Select one or more records, and click the Do Not Include Associations link. A red asterisk appears next to the record name in the Record Data panel for any records that will not include association information during the migration process. To reinclude record association information, select one or more records in the Record Data panel, and click the Include Associations link.</p> <p>For those packages generated with older versions, export your packages again.</p>
Section	Business object sections are merged in the Object Migration import.
Section field	Section fields are merged in the Object Migration import.
Security group	<p>All security group information is overwritten by the import process.</p> <p>Security group members are not included in the import process.</p>
State family	The Object Migration import overwrites business object state families.
Static list	All static list types are overwritten by the import process. However, static list values are merged.
Structure	The import process overwrites structure.
Style	Object Migration's import tries to replace styles, with 2 exceptions: System styles and styles that are in use are not replaced.

Table 8. Object migration import tips (continued)

Workflows	<p>Workflows are overwritten by an import, but the version is retained and the state of the workflow is preserved.</p> <p>When you import a workflow and the workflow is triggered by an association with an association string that is not currently in the Association Types list, the import adds the string to the Association Types list.</p> <p>Asynchronous Workflows (such as action, association, de-association)</p> <ul style="list-style-type: none">• Copy and rename the workflow• Retire the original workflow• Activate the copied workflow <p>Synchronous Workflows</p> <ul style="list-style-type: none">• Rename the workflow
-----------	--

Chapter 4. Migrating database table indexes

You can view, alter, drop, and create database table indexes in the Database Table Manager tool. In addition, the Database Table Manager supports the migration of database table indexes from one IBM TRIRIGA environment to another.

About this task

The process of migrating database table indexes is done through the migration of record data on the triPlatformDBTableManager module.

The Database Table Manager tool is not found in the IBM TRIRIGA Administrator Console. With the proper permissions, you can find the Database Table Manager by signing into IBM TRIRIGA and navigating to **Tools > System Setup > System > Database Table Manager**.

Exporting database table indexes

You can create an export package that either exports all indexes of a table or exports individual indexes.

About this task

Note: In terms of globalization, the records that are created from certain Database Table Manager business objects, triIndex, triSparseColumn, triTable, and triIndexedColumn, are not supposed to be translated. However, because their BO name mappings contain a list field, the platform automatically attempts to export the data for translation. To prevent this process, the records of these specific business objects are always excluded during data export from Globalization Manager.

Exporting all indexes of a table

You can select a table for package inclusion and all of the indexes that are part of the table are automatically included in the package.

Procedure

1. When you build the export package in the **Object Finder** panel of the Object Migration tool, specify **Record Data** for the **Object Type** search parameter.
2. For **Module**, select **triPlatformDBTableManager**.
3. For **Business Object**, select **triTable** and click **Search**.
4. Search for the table that contains the indexes that you want to add.
5. Select the table.

All indexes that are associated with the table are added to the export package.

Exporting individual indexes

You can select individual database table indexes to include in your export package.

Procedure

1. When you build the export package in the **Object Finder** panel of the Object Migration tool, specify **Record Data** for the **Object Type** search parameter.
2. For **Module**, select **triPlatformDBTableManager**.
3. For **Business Object**, select **triTable** and select **Search**.
4. Search for the table that contains the indexes that you want to add.

5. Select **Find Dependents** for the selected table. All of the indexes for that table appear as associated Record Data.
6. Select the individual indexes to include in the package.

Chapter 5. Configuring the Object Migration log

The export and import processes in Object Migration generate informational messages when an object's dependent object names cannot be determined or verified. The messages are sent to a log file named `ObjectMigration.log`.

About this task

The messages in the log identify the name of the object that has an issue and lists its invalid references by type and where they are referenced. The log is in the log directory of your IBM TRIRIGA installation, for example, at `C:\Tririga\log\ ObjectMigration.log`.

You configure the `ObjectMigration.log` file in the `log4j.xml` file in the `config` directory of your installation, for example, at `C:\Tririga\config\log4j.xml`. Be sure to review these settings and edit them as you want before you do Object Migration. Also, it is important to review the `ObjectMigration.log` after each Object Migration event.

The default Object Migration entries in `log4j.xml` follow.

```
=====  
Append messages to the ObjectMigration.log file  
=====  
<appender name="OM" class="org.apache.log4j.DailyRollingFileAppender">  
  <param name="File" value="C:/Tririga/log/ObjectMigration.log"/>  
  <param name="Append" value="true"/>  
  <param name="Threshold" value="INFO"/>  
  <param name="DatePattern" value="'.'yyyy-MM-dd"/>  
  
  <layout class="org.apache.log4j.PatternLayout">  
    <param name="ConversionPattern" value="%d %-5p %m%n"/>  
  </layout>  
</appender>  
...  
  
<  
| Additivity is set to false so OM will not log with the root elements.  
| This allows only OM to log in the ObjectMigration.log file.  
-->  
<category name="com.tririga.om.logger.OMLogger" additivity="false">  
  <appender-ref ref="OM"/>  
</category>  
<category name="com.tririga.platform.om.logger.OMLogger" additivity="false">  
  <priority value="INFO"/> <!-- Set to DEBUG to enable logging -->  
  <appender-ref ref="OM"/>  
</category>
```

`com.tririga.om.logger.OMLogger` is for versions 2.1 to 2.5 of Object Migration.

`com.tririga.platform.om.logger.OMLogger` is for versions 2.6+ of Object Migration.

Chapter 6. Exporting workflows as text

A workflow can be exported to a text file. This text file contains a text-based model of the workflow logic. You can export a workflow before you include it in a migration package, or before you import a migration package that contains a workflow.

About this task

You might find it useful to export a workflow that is contained in a migration package to examine it before you decide to import it. You might want to compare a workflow that is contained in a migration package with another version of the same workflow by using a comparison tool to identify differences between versions.

1. [Upload](#) the Object Migration package that contains the workflow into the target platform environment.
2. [Validate](#) the objects that are to be migrated.
3. Select one or more workflows in the **Workflow** panel and click **Text Export Selected**.
4. Save the file to the system.

Individual workflows are saved with a `.cld` file extension. If you select multiple workflows to export as text, the workflows are saved in a `.zip` file. The `.zip` file includes a folder hierarchy that is based on the object structure of the workflow. The hierarchy includes the workflow module and business object, if defined.

5. Use a text editor to examine the exported workflow and verify that it should be imported as part of the migration package.
6. [Import](#) the migrated objects into the target environment.

Chapter 7. Downloading UX content

UX content can be downloaded as one or more content files, such as .html and .css files. You can download a content file before you include it in a migration package, or before you import a migration package that contains a content file.

About this task

You might find it useful to download a content file that is contained in a migration package to examine it before you decide to import it. You might want to compare a content file that is contained in a migration package with another version of the same content file by using a comparison tool to identify differences between versions. For more information on comparing UX metadata, see [Compare and Merge HTML Views](#) on the [IBM TRIRIGA wiki](#).

1. [Upload](#) the Object Migration package that contains the UX content into the target platform environment.
2. [Validate](#) the objects that are to be migrated.
3. Select **Web Component** or **Web View File** in the **Navigation** panel.
4. Select one or more UX components or content files, respectively, in the **Objects** panel and click **Download Content For Selected**.
5. Save the files to the system.

The downloaded .zip file includes the OM folder and System folder. The OM folder contains the source content files, while the System folder contains the target content files.

6. Use an HTML "diff" (difference) tool to examine the downloaded content and verify that it should be imported as part of the migration package.
7. [Import](#) the migrated objects into the target environment.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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

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

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

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

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan*

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 jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

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

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

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.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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.

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

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

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (your company name) (year).

Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. _enter the year or years_.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux® is a trademark of Linus Torvalds 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.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other product and service names might be trademarks of IBM or other companies.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM’s Privacy Policy at <http://www.ibm.com/privacy> and IBM's Online Privacy Statement at <https://www.ibm.com/privacy/details/us/en/> in the section entitled “Cookies, Web Beacons and Other Technologies.”



Part Number:

(1P) P/N: