

IBM TRIRIGA Application Platform
Version 3 Release 5.2

Globalization User Guide



Note

Before using this information and the product it supports, read the information in "Notices" on page 47.

This edition applies to version 3, release 5, modification 2 of IBM TRIRIGA Application Platform and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 2011, 2016.**

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

Contents

| | |
|---|-----------|
| Chapter 1. Globalizing applications | 1 |
| Options for globalization | 1 |
| Tasks to support globalization | 1 |
| Chapter 2. Importing language packs | 3 |
| Language packs | 3 |
| Creating language codes | 4 |
| Finding imported languages | 4 |
| Importing language packs for data and labels | 4 |
| Chapter 3. Globalizing record data | 7 |
| Internal values. | 7 |
| Translated values. | 8 |
| Changing the base language | 8 |
| Globalizing business objects by using the Data Modeler | 9 |
| Setting business object fields as localizable | 9 |
| Fields that cannot be localized by using Data Modeler | 9 |
| Object migration. | 11 |
| Publication of business objects | 11 |
| Publication of language tables | 12 |
| Importing localizable data by using the integration object | 13 |
| Importing localizable data by using DataConnect. | 13 |
| Runtime usage of globalized data | 15 |
| Runtime usage of globalized data in forms | 15 |
| Runtime usage of globalized data in workflow mappings | 16 |
| Runtime usage of globalized data in reports | 17 |
| My Reports | 21 |
| Community Reports | 21 |
| Chapter 4. Exporting data and labels for translation. | 23 |
| Exporting labels and record data for translation | 23 |
| Export of database table indexes | 23 |
| Storage of translation files | 24 |
| Chapter 5. Translating labels directly | 27 |
| Chapter 6. Globalizing lists | 29 |
| Types of lists | 29 |
| Specifying list items in alternative languages | 29 |
| Chapter 7. Globalizing date and time | 31 |
| Custom date and time formats | 31 |
| Defining Date or Date and Time Format. | 32 |
| Chapter 8. Globalizing currency | 33 |
| Currency | 33 |
| Creating currency conversion rates | 34 |
| Chapter 9. Globalizing units of measure | 35 |
| Rounding rules for UOM values | 35 |
| Creating UOM types | 36 |
| Creating UOM values | 36 |
| Chapter 10. Globalizing BIRT reports | 37 |
| Dates | 37 |
| Translations | 37 |
| Chapter 11. Setting language preferences in profiles | 39 |
| Chapter 12. Globalizing the Gantt scheduler | 41 |
| Changing the Gantt language | 41 |
| Chapter 13. Troubleshooting globalization | 43 |
| Troubleshooting import of language packs | 43 |
| Troubleshooting record data globalization | 44 |
| Troubleshooting UX globalization | 45 |
| Notices | 47 |
| Trademarks | 49 |
| Terms and conditions for product documentation. | 49 |
| IBM Online Privacy Statement | 50 |

Chapter 1. Globalizing applications

The business environment today is increasingly interconnected. Your customers and employees might come from multiple countries and regions, and might speak multiple languages. You can use the IBM® TRIRIGA® Application Platform to display text and data in other languages. You can also customize data, labels, units of measure, dates, and currency for multicultural environments.

Options for globalization

Globalization, also known as localization or multicultural support, is the process of organizing an application so that it can be configured to work in different countries and in different languages.

Globalization can raise logistical issues that relate to workflow behavior in different languages or countries. For example, business logic for an application can vary because of different regulations. The globalization requirements of your applications might vary. However, the following options are the most common:

US English

Your applications are used only in US English.

One language, not US English

Your organization conducts its business in one official language and presents information to users only in that language.

Official language with multilingual workforce

Your organization conducts its internal business in one official language but presents information to users in their own languages.

Multilingual with no official language

Your company does not have an official language and users work with data in multiple languages.

All business objects that come with the IBM TRIRIGA Application Platform are written in US English and use units of measure and currency that are appropriate for the United States. If you use the US English option, you do not need to globalize your applications unless they were developed in another language. If you use one of the other options, you must globalize your applications to support the required languages.

You can use globalization features to adjust an application to an organization. For example, some labels or descriptions are supplied with an application but your organization decides that they do not suit its business practices. Instead of directly modifying any forms, you can define a language and supply translations for specific labels and descriptions to match business practices. If you customize an application this way, you can avoid incompatibilities between modified business objects and business objects.

Tasks to support globalization

If your users operate in a language that is not US English, you must ensure that their required information is supported.

You can use the following checklist to plan for globalization.

- Identify the base language and the secondary languages that are supported. If necessary, change the base language from US English.
- If the languages are supported by the IBM TRIRIGA Application Platform, obtain and import the language packs. Otherwise, add the languages to the language list, create the language codes, and configure the locales in the Globalization Manager.

The language packs are in the `tririga_version_date_language_packs.zip` file on the Application Upgrade image in the IBM TRIRIGA application eAssembly on the IBM® Passport Advantage® website.

After you download the installation image that contains the `tririga_version_date_language_packs.zip` file, you must add the file to the `<install_directory>\userfiles\LanguagePacks` folder.

- Translate record data. You can use the Globalization Manager, the integration object, or DataConnect to import record data. Or you can log in as a user of the supported language and edit the localizable fields in the forms.
- Use the Globalization Manager to translate labels, tab information, system messages, and tooltips.

If you choose to do a delta export, the `tririga_version_date_language_packs.zip` file must be in the `<install_directory>\userfiles\LanguagePacks` folder when you do the export.

- Use the List Manager to translate lists.
- Use the **Tools** menu to translate approvals, CTQ dynamic lists, news, notification content, offline content, and user message records into the supported languages.
- If none of the date or time formats that are included with the IBM TRIRIGA Application Platform are appropriate for your business environment, add the formats in the List Manager.
- Add the currency type to the List Manager if needed. To avoid data loss or other conversion issues, add, or remove currencies before you add the data records.
- Translate units of measure and UOM values in the **Administration** menu if needed.
- If you have an official language, translate all reports into that official language. If you need reports in multiple languages, create multiple versions of the reports. When you create these reports, you can choose to filter data by the `triLanguageLI` field so your reports do not mix data in multiple languages.
- Translate all help text into supported languages.
- Establish language, currency, time zone, date format, date time format, area unit, and length unit values in the My Profile pages.
- Enable bidirectional support and specify the direction of user-generated text in the My Profile page if needed.
- To use a language with multi-byte characters, such as Japanese or Russian, you might need to configure your database. Check the installation checklist for your database. To use an Excel file with multi-byte characters, open the file with a text editor and save it as UTF-8.

Chapter 2. Importing language packs

Language packs contain the translated labels and data that support the use of different languages. You use the Globalization Manager to import language packs for record data and for labels.

Language packs

You use the language packs when you want to change the base language or to support secondary languages.

To use the language packs, you must first obtain the `tririga_version_date_language_packs.zip` file. The file is on the Application Upgrade image in the IBM TRIRIGA application eAssembly on the IBM® Passport Advantage® website.

After you download the installation image that contains the compressed language packs file, you must add the file to the `<install directory>\userfiles\LanguagePacks` folder.

The following sample shows the structure of the compressed file:

```
tririga_version_date_language_packs.zip
- ar.zip (Arabic)
  - data
    -ar.zip
    - *.xliff
  - label
    -ar.zip
    - *.xliff
- *.zip
  - data
    -*.zip
    - *.xliff
  - label
    -*.zip
    - *.xliff
- ...
- guitooltips
  - GUITooltips_US_English_date.zip (US English)
```

You must unpack the main language pack, such as `tririga_version_date_language_packs.zip` before you select the individual language pack to import, such as `ru.zip`. Each individual language pack contains a data language pack in the `data` directory and a label language pack in the `label` directory.

In the data language pack, record data is grouped by business object in the format `Module Name_BO Name.xliff`, for example, `Classification_triAreaType.xliff`. Each data language pack contains a maximum of 100 records. If more than 100 records exist for a business object, extra data language pack files are created, for example `Classification_triAreaType_01.xliff`, `Classification_triAreaType_02.xliff`.

In the label language pack, files are grouped by label type in the format `Module Name_LabelType.xliff`, for example, `Document_GUILABEL.xliff`. Each label file contains a maximum of 1000 labels. If more than 1000 labels exist for a label type,

extra label language pack files are created, for example Document_GUILABEL_01.xliff, Document_GUILABEL_02.xliff.

Importing language packs does not modify the internal values of the labels and record data that are included with IBM TRIRIGA.

Creating language codes

IBM TRIRIGA includes a standard set of languages that are specified by language and country codes. If your language is not included, you can add the language to the list.

Procedure

1. Select **Tools > Administration > Lists**.
2. In the **Manage By** section, select **Language**.
3. In the **Value** field, enter the name of the language, and click **Save Entries**. For example, enter Brazilian Portuguese.
4. Use the Administrator Console to refresh the Object MetaData cache.
5. Configure the locale in the Globalization Manager by selecting **Tools > Administration > Globalization Manager** and clicking **Language Code**. Language codes are in the following format: *language_COUNTRY*. For example, the language code for Brazilian Portuguese is pt_BR.

Finding imported languages

If you have administrative access, you can use the Administrator Console to find information about the language packs that are imported into IBM TRIRIGA. This information includes the name of the language packs, when the packs were imported, and by whom.

Procedure

1. Sign in to the Administrator Console.
2. In the Managed Objects pane, select **Database Query Tool**.
3. In the **SQL Query** field, enter the following SQL statement:

```
select * from ENVIRONMENT_PROPERTIES
where environment in (
'platform.any.lang.file.import',
'platform.data.lang.swap.import',
'platform.data.lang.secondary.import',
'platform.label.lang.primary.labelImport',
'platform.label.lang.import')
```
4. Click **Run Query**.

Importing language packs for data and labels

You use the Globalization Manager to import language packs for labels or record data. You can import XLIFF files in ZIP format. To import single XLIFF files, first zip them.

Before you begin

You must use the language packs that are included with the current IBM TRIRIGA application.

Earlier or customized versions of IBM TRIRIGA might lack some translations.

About this task

Even if only a few language files for select data are imported to the system, the system calculates all of the localized record paths in the system.

Procedure

1. Select **Tools > Administration > Globalization Manager** and click **Import**.
2. Select whether you want to update the existing translation. If you select **Yes**, customized or existing translations might be overwritten.
3. Select the language packs to import.
4. Click **Import**.

Results

The translated data and labels are created or updated based on the information that is contained in the files that you import.

Chapter 3. Globalizing record data

By using the language packs that are delivered with IBM TRIRIGA, you can globalize record data for your multicultural users. Record data includes internal and translated values.

Internal values

Internal values are used when consistent or literal strings are required, for example, in a workflow condition, a static filter in a system report, or a formula. Internal values are stored in the main storage table, also named the T_ table.

The internal value determines the value that is displayed in forms and reports when a translation is not available for the user language. For example, the internal values are written in Brazilian Portuguese and French is a secondary language. A French user views a record, and the description field does not have a French translation. The Brazilian Portuguese value of the field is displayed. In another example, a French user views a field that does not have French translation, and the value of this field was included with IBM TRIRIGA. The text displays the internal value which is created in the default language, US English.

The internal values of the application records that are included with IBM TRIRIGA are in US English, which is the default base language. If a record is created, the first value is stored in the main storage table. Subsequent values are stored in the main storage or language tables according to the rules outlined in the following table:

Table 1. Translated data in forms

| Base language | User language | Data storage and display |
|----------------|---------------------------------------|--|
| US English | US English | The main table (T_). |
| US English | Not US English (for example, Italian) | The language table (L_) where the LANGUAGE_ID column corresponds to the language ID of the user. |
| Not US English | US English | The language table (L_) where the LANGUAGE_ID column is 1. |
| Not US English | Not US English | The language table (L_) where the LANGUAGE_ID column corresponds to the language ID of the user. |

Users who are not administrators can update internal values only if the base language and the user language are US English.

Administrator users can also import internal values by using the following tools:

Object Migration (OM) tool

When the OM package contains records, and those records are imported using the OM tool, the values of those records are saved to the T_ tables. Values that are imported from the OM package become the internal values regardless of the base language or the user language.

DataConnect

When the DC_JOB process is run by a US English user and the base language is US English, the data is imported into internal values.

Translated values

Translated values are stored in the language tables, also called the L_ tables. If the base language is not US English, then translated values include base language values. Secondary language values are always included in translated values.

Base values are the localizable data that is stored in the language tables and that are assumed to be written in the base language. By default, the base language is US English. If the base language remains in US English, internal values are base language values. Otherwise, the base language values are stored in the language tables with other language values.

Secondary languages are languages other than the base language. Secondary language values include the different translations of the localizable data that is written in the supported secondary languages and that is stored in the language tables. Secondary language values are always stored in the language tables. You can add secondary languages in the Globalization Manager.

If the base language is not US English, US English is treated as a secondary language, and US English values are stored in the language table.

Changing the base language

You use the Globalization Manager to change the base language. The default language is US English. If you change the base language from the default, US English users become secondary language users and cannot change the internal values. Changing the base language does not affect the translated data or the language values that are already imported.

Before you begin

Ensure that the language pack of the new base language is imported into IBM TRIRIGA.

Procedure

1. Select **Tools > Administration > Globalization Manager** and click **Base Language**.
2. Select the target base language.
3. Click **Change**.

What to do next

If your system has multiple process servers, restart the IBM TRIRIGA server.

You can revert the base language back to US English. Reverting the base language does not update the imported language values.

Globalizing business objects by using the Data Modeler

As an application developer, you use the Data Modeler to globalize your business objects. The Data Modeler uses an object-oriented approach to define and maintain the structures of IBM TRIRIGA data.

Setting business object fields as localizable

In the Data Modeler, you can support localizable values for text and note fields. If a field is set to localizable, it can have a secondary language other than US English.

About this task

If you upgrade IBM TRIRIGA Application Platform without upgrading the included applications, you must manually update the **Localizable** property, especially for your custom applications.

If you clear the **Localizable** property, the column is removed from the language table when you publish the business object, and data might be lost.

If you set a field to localizable, and another business object refers to the field as a smart section or locator, the related fields in the smart section or locator automatically become localizable. When you publish the main business object, the related business objects are republished and language columns are created for the related fields.

Procedure

1. Select **Tools > Builder Tools > Data Modeler** and revise the business object.
2. Open the field that you want to support globalization. On the Property page, in the General section, select the **Localizable** property. Save the field.
3. Repeat step 2 for each field that you want to support globalization.
4. Save the business object.

What to do next

Publish the business object.

Fields that cannot be localized by using Data Modeler

Data Modeler cannot be used to localize all fields and business objects. For example, Data Modeler cannot localize locator fields, project names, or smart sections with live links. You cannot select the **Localizable** property for these fields. Instead, the value of **Localizable** property is derived from other sources.

Table 2. Fields that cannot be localized from Data Modeler

| Field type | Source of Localizable Property | What happens during business object publication if field is localizable? | Source of localized data |
|---------------------------------|--|---|--|
| Locator text | Source field of the locator field. For example, the triNameTX field of the triFloor business object is the source field of the triParentFloorTX locator field of the triSpace business object. | A column is added into the language table, for example, the triParentFloorTX column of the L_triSpace language table. | Language table of the locator field, for example, L_triSpace. |
| Record name (system read-only) | Business object. If one of the fields in the business object name mapping is localizable, the name field is also localizable. | A static column, SPEC_NAME, is created in the language table. | SPEC_NAME column of language table. |
| Path (system read-only) | <ul style="list-style-type: none"> • Non-hierarchical module: Its business object name mapping. For example, the triIdTX field of the triCapitalProject business object. • Hierarchical module: Its base business object name mapping. For example, the triNameTX field of the Geography base business object. | A static column, SYS_PATH, is created in the language table. | SYS_PATH column of language table. |
| Project name (system read-only) | Project business object Name mapping | No column is created in the language table. | SPEC_NAME column of the language table of the project record. |
| Classification | Root classification business object name mapping | No column is created in the language table. | SPEC_NAME column of the language table of the classification record. |
| Business object | Business object name mapping | No column is created in the language table. | SPEC_NAME column of the language table of the business object. |
| Smart section – Live link | Source field | No language table is created for this type of smart section. | Language table of the referenced or linked record. |

Table 2. Fields that cannot be localized from Data Modeler (continued)

| Field type | Source of Localizable Property | What happens during business object publication if field is localizable? | Source of localized data |
|-------------------------------|--------------------------------|--|---|
| Smart section – Non-live link | Source field | A language table is created for this type of smart section. | Intermediate or smart section language table. |

Object migration

The Localizable property of a business object field is supported in object migration (OM). OM packages that were created before IBM TRIRIGA Application Platform 3.3 override or reset the localizable property of the target environment.

When an object migration package that was created before IBM TRIRIGA Application Platform 3.3 is imported into an environment that was upgraded to IBM TRIRIGA Application Platform 3.3 or later and IBM TRIRIGA 10.3 or later, the language table is deleted. The localizable property does not exist in an object migration package that was created before IBM TRIRIGA Application Platform 3.3. As a result, the localizable property value that is already in the target system is reset. By design, the business object publish deletes the column for a field that is no longer marked as localizable. Since an old object migration package does not have this property, all the columns are dropped, which drops the whole language table.

Publication of business objects

When you publish a business object that contains localizable fields, language tables are created to store the translated values. If more fields are localizable, the same language table is updated to accommodate the additional localizable fields.

Reference fields rely on the localizable properties of source fields. When you publish a source business object, the referenced business object is also published. The localizable property of the source field is then associated with the reference field.

For example, the triParentFloorTX locator field in the triSpace business object references the triNameTX field in the triFloor business object. When you publish the triFloor business object, the triSpace business object is also published, and language columns are created in both business objects.

To investigate problems with your dependencies, you can determine dependencies by using the following methods:

- Use the Where Used tool to find dependencies, and publish the main business object to resolve these dependencies.
- If you cannot resolve all dependencies, publish all business objects in the system. The scripts for publishing business objects are located in the \Scripts folder of the upgrade disk. For Oracle and IBM DB2® databases, run the Publish_all_BOs_For_localization_ORCL.sql script. For Microsoft SQL Server databases, run the Publish_all_BOs_For_localization_MSSS.sql script.

Publication of language tables

When you publish a business object, the language table is given a similar name to the main table, but with an L_ prefix instead of a T_ prefix. For example, if the main table is T_TRIAREATYPE, the language table is L_TRIAREATYPE.

The field or column names in the language table are the same as the fields or columns in the corresponding main table, which is shown in the following graphic:

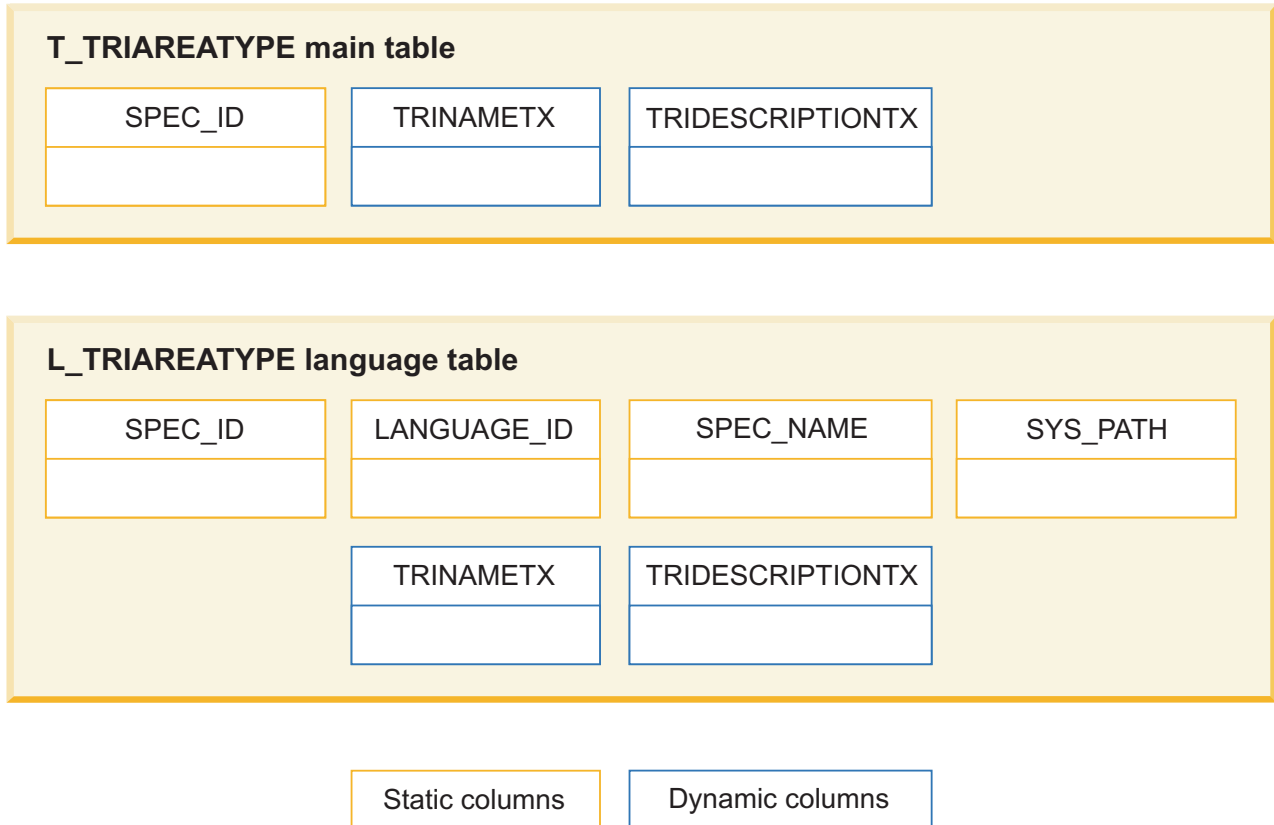


Figure 1. Example of a main table that is mapped to a language table for a business object

Language tables are also created for smart sections that do not contain live links. For example, if the triTask business object has a smart section that is named triFacilitiesProject, the following language table is created:

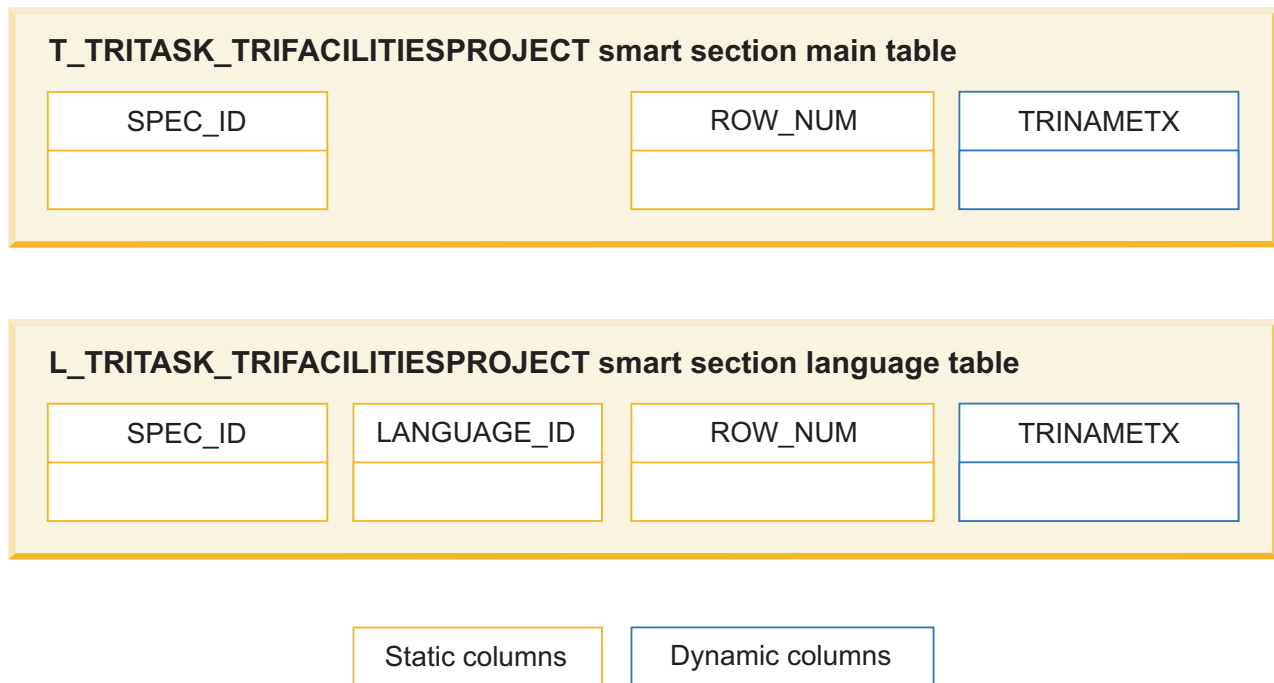


Figure 2. Example of a main table that is mapped to a language table for a smart section

Language values for note fields are stored in the static database table that is named L_IBS_SPEC_VALUE_NOTES. The localizable property is also supported in the Object Migration (OM) tool.

Importing localizable data by using the integration object

The integration object uses database tables, files, and staging tables to import localizable data into IBM TRIRIGA. In the integration object, when you execute the integration, the user language in your profile determines the language that is used to store the imported data.

Importing localizable data by using DataConnect

You can use DataConnect to create instances or translated records for your business objects. DataConnect supports the import of localizable data through the DC_JOB tables, which you can access through the integration object or externally. You must include all data in a record when you use DC_JOB to import globalized data.

About this task

The language of the user in the DC_JOB table determines where the data is stored. If you use the integration object to import the DataConnect tables, the DC_JOB user is the person who executes the integration. If you run DataConnect outside the integration object, you must manually specify the user in the DC_JOB table.

Example

Two S_TESTDC staging tables are used, one for US English data and one for French data. The first DataConnect job imports US English data into the main

table.

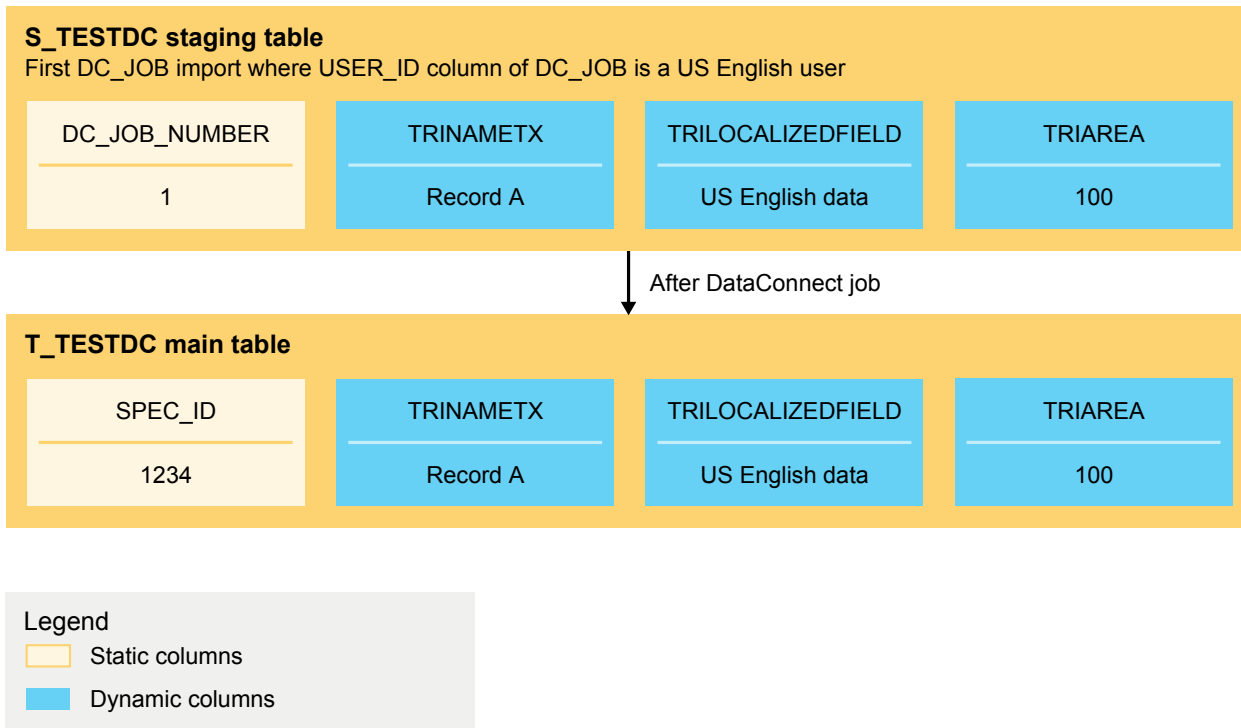


Figure 3. After importing one DataConnect job with localizable data

The second DataConnect job imports French data into the language table. When both DataConnect jobs are complete, the T_TESTDC main table has a row for US English, and the L_TESTDC language table has a row for French.

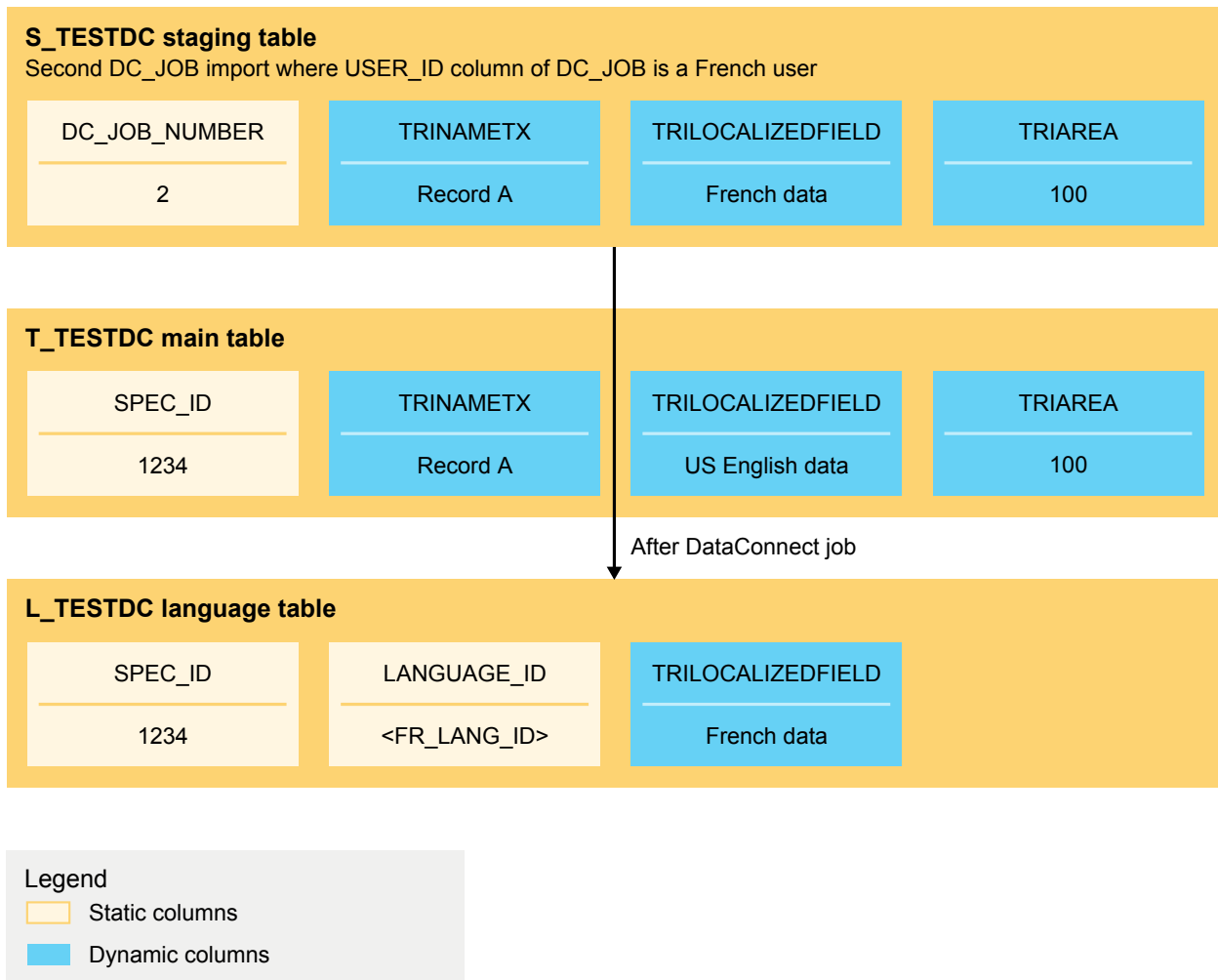


Figure 4. After importing two DataConnect jobs with localizable data

If a user later imports data for another language that is not US English or French, another row is added to the L_TESTDC table. For example, if the language of the user in the DC_JOB is Italian, a row is added for Italian data.

Runtime usage of globalized data

When the business objects are published, globalized data is displayed in IBM TRIRIGA record forms, workflow mappings, and reports.

Runtime usage of globalized data in forms

IBM TRIRIGA uses the user language to render and process localizable field values in the record forms.

Base language users

For base language users, the IBM TRIRIGA Application Platform renders the base values that come from the language tables in the user language.

If the base language of the system remains the default US English, then all base language users or US English users persist their values to the main tables and not to the language tables.

User updates are persisted in the language tables. These updates do not overwrite the internal value from the main tables. If the internal value is null or empty, the value is stored in both main and language tables.

Secondary language users

For secondary language users, IBM TRIRIGA Application Platform renders the language values that come from the language tables. If no language values are available in the user language, the internal values are displayed.

If a user edits a field in a form that contains the internal value, this internal value is persisted to the language table. User updates are persisted in the language tables. Validation of required fields is applied on internal values and on secondary values.

Internal, base, and other language values are not affected by changes to the secondary language value. If the base language is not US English, US English users are considered secondary users and their values are persisted in the language tables.

Runtime usage of globalized data in workflow mappings

If a data language pack is imported, globalized data is supported in workflow mappings.

The following mappings are supported:

- Localizable text field to localizable text field
- Localizable text field to localizable locator field
- Localizable text field to a localizable note field
- Record to localizable locator field
- Literal string to localizable text field
- Localizable classification field to a localizable text field
- System form label translation to a localizable text field

If the value of the localizable field is mapped to a record, its internal value, base value, and language values are also copied to the target record.

At run time, workflows map all available globalized data from the source field to the target field. If globalized data is available in the source record but not the target record, a new globalized record is created in the target record. If no equivalent localizable value exists in the source record, an empty value is set in the target record. An example is displayed in the following graphic.

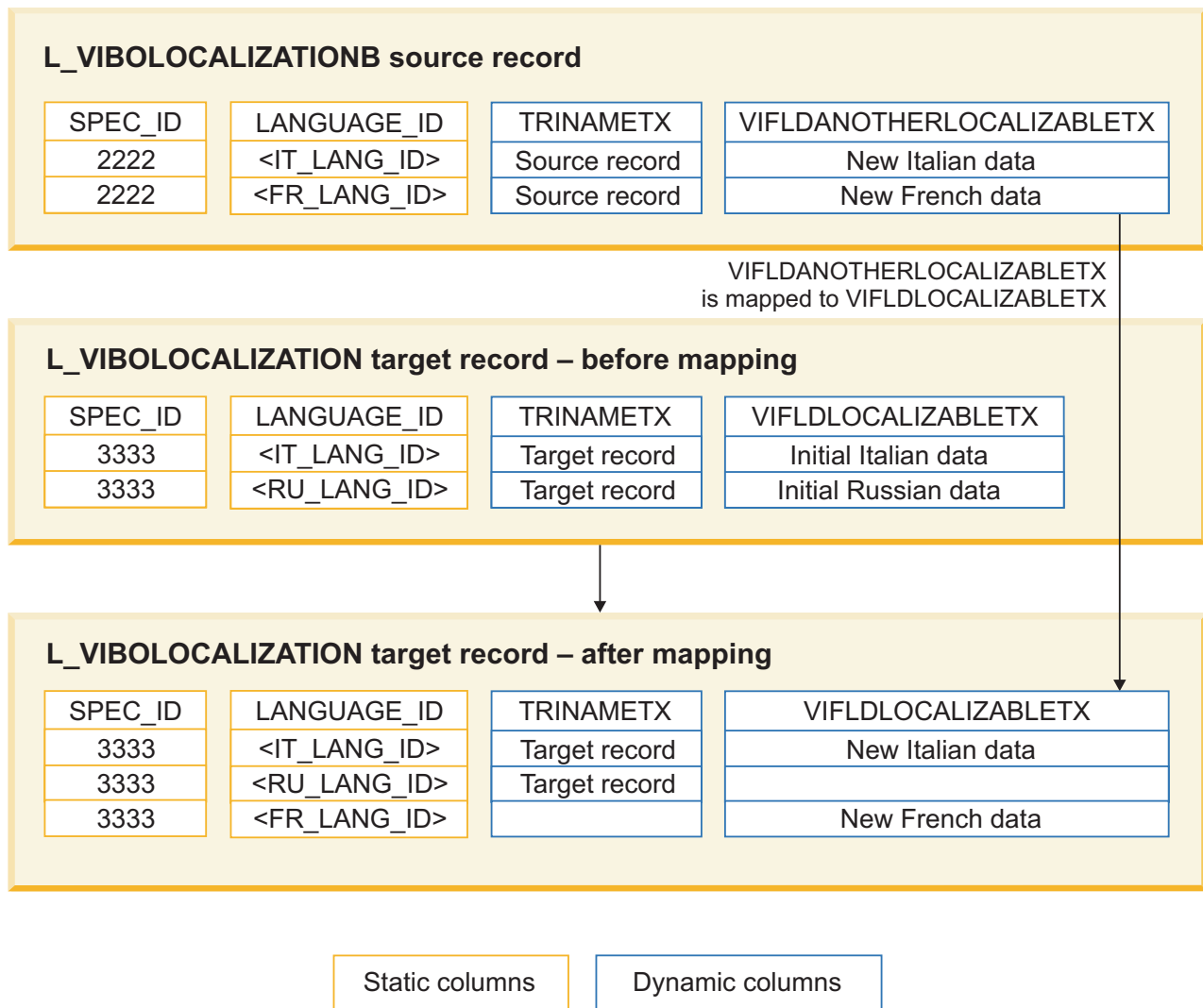


Figure 5. Mapping in workflow run time

At run time, workflows can use literal string values, dynamic values, and extended formulas. The internal value of a localizable field from the main tables is always used in the workflow engine.

Runtime usage of globalized data in reports

The report engine queries and displays values of localizable fields in the user language. Non-localizable fields are queried from the main storage tables. Localizable record data is displayed in the language of the user. You can run reports in the Report Manager, report sections in forms, and portal sections.

If the base language of the system remains the default US English, then the report engine queries, filters, and sorts values from the main tables and not from the language tables, for all base language users or US English users.

For base language users, the report engine queries, filters, and sorts localizable values from the language tables in the base language. For secondary language users, the report engine queries, filters, and sorts localizable values from the language tables.

If the user updates an editable report, the same rules are applied to the form changes at run time. Validation of required fields is applied on secondary values.

The report engine treats values that are inputted into the user filter at run time to be in the preferred language of the user. The report engine uses these values to filter from language tables for base or secondary language users.

If the **Localize** flag is enabled for a system filter value in the System Filter Columns section of the report, the report engine uses that value to filter the results, based on the language of the user at run time. If you create a report in a language other than US English, you can use the **Localize** flag to specify that the report runs only for users of your language. If you want the report to be run in multiple languages, you do not use the **Localize** flag, and the value of the system filter must be an internal value.

Note: For scenarios where the filter is run against the language table, if there is no language value that is available for that language, then the internal value in the main table is considered when filtering the results.

Example 1: Base language is US English

The base language of the system is US English. The triPeople report filters the triStatusCL field to be equal to Retired in US English and Retir   in French. The report engine compares the system filter with the records that are selected in the report, as in the following table.

Table 3. Example of report engine comparisons when the base language of the system is US English

| System filter value | Localize flag selected? | User language at run time | What does the report engine do at run time? |
|-------------------------|-------------------------|---------------------------|---|
| Retired (in US English) | No | Any user language | Considers system filter value as internal value, and filters results based on the matching value from the main table (T_ table). |
| Retired (in US English) | Yes | US English | Considers system filter value as in the language of the user at run time. Because the base language is US English, the system filter value is considered as internal value, and filters results based on the matching value from the main table (T_ table). |
| Retired (in US English) | Yes | Not US English | Considers system filter value as in the language of the user at run time, and filters results based on the matching value from the language table (L_ table), where LANGUAGE_ID column matches the language of the user at run time. Note: This is not likely to return any results for non-US English users unless there are "Retired" values in the language table for their respective language. |

Table 3. Example of report engine comparisons when the base language of the system is US English (continued)

| System filter value | Localize flag selected? | User language at run time | What does the report engine do at run time? |
|---------------------|-------------------------|---------------------------|--|
| Retir  (in French) | No | Any user language | Considers system filter value as internal value, and filters results based on the matching value from the main table (T_ table). Note: This is not likely to return any results for any users unless there are "Retir " values in the main table (T_ table). |
| Retir  (in French) | Yes | US English | Considers system filter value as in the language of the user at run time. Because the base language is US English, the system filter value is considered as internal value, and filters results based on the matching value from the main table (T_ table) Note: This is not likely to return any results for US English users unless there are "Retir " values in the main table (T_ table). |
| Retir  (in French) | Yes | Not US English | Considers system filter value as in the language of the user at run time, and filters results based on the matching value from the language table (L_ table), where LANGUAGE_ID column matches the language of the user at run time. Note: This is not likely to return any results for non-French users unless there are "Retir " values in the language table for their respective language. |

Example 2: Base language is non-US English

The base language of the system is non-US English. The triPeople report filters the triStatusCL field to be equal to Retired in US English and Retir  in French. The report engine compares the system filter with the records that are selected in the report, as in the following table.

Table 4. Example of report engine comparisons when the base language of the system is non-US English

| System filter value | Localize flag selected? | User language at run time | What does the report engine do at run time? |
|-------------------------|-------------------------|---------------------------|--|
| Retired (in US English) | No | Any user language | Considers system filter value as internal value, and filters results based on the matching value from the main table (T_ table). |

Table 4. Example of report engine comparisons when the base language of the system is non-US English (continued)

| System filter value | Localize flag selected? | User language at run time | What does the report engine do at run time? |
|-------------------------|-------------------------|---------------------------|---|
| Retired (in US English) | Yes | US English | <p>Considers system filter value as in the language of the user at run time, and filters results based on the matching value from the language table (L_ table), where LANGUAGE_ID column is US English language.</p> <p>Because the base language is not US English, the US English users are considered as any other language users in the system, so the report engine is filtering the values from the language table.</p> |
| Retired (in US English) | Yes | Not US English | <p>Considers system filter value as in the language of the user at run time, and filters results based on the matching value from the language table (L_ table), where LANGUAGE_ID column matches the language of the user at run time.</p> <p>Note: This is not likely to return any results for non-US English users unless there are "Retired" values in the language table for their respective language.</p> |
| Retir  (in French) | No | Any user language | <p>Considers system filter value as internal value, and filters results based on the matching value from the main table (T_ table).</p> <p>Note: This is not likely to return any results for any users unless there are "Retir " values in the main table (T_ table).</p> |
| Retir  (in French) | Yes | US English | <p>Considers system filter value as in the language of the user at run time, and filters results based on the matching value from the language table (L_ table), where LANGUAGE_ID column is US English language.</p> <p>Because the base language is not US English, the US English users are considered as any other language users in the system, so the report engine is filtering the values from the language table.</p> <p>Note: This is not likely to return any results for US English users unless there are "Retir " values in the language table for US English users.</p> |
| Retir  (in French) | Yes | Not US English | <p>Considers system filter value as in the language of the user at run time, and filters results based on the matching value from the language table (L_ table), where LANGUAGE_ID column matches the language of the user at run time.</p> <p>Note: This is not likely to return any results for non-French users unless there are "Retir " values in the language table for their respective language.</p> |

My Reports

Similar to a user filter in the User Filter Columns section of the report, a system filter input by the owner of the report is treated in the language of the owner if the **Localize** flag is checked. By default, when a system filter is added, the **Localize** flag value is checked. If the flag value is cleared, the system filter does not filter against the language table even if the filter is in My Reports.

Community Reports

Depending on how you copy or share community reports, you might be required to update the system filter value.

From My Reports

When a report is copied as a community report, the platform creates a separate template of this report that includes the system filter value in the community page and the report is marked as read-only. This copy becomes a community report.

The system filter value is now considered as in the language of the user who is running the report if the **Localize** flag is checked. This might not yield the desired results if the language of the user who is running the report is not the same as the language of the owner who originally created the report. In this case, the community user needs to copy the community report to his My Reports page and update the system filter value to be in his language.

From System Reports

When a system report is shared to the community, the platform does not create a new instance of the report. It is the same system report when run from the community page. The query engine treats the system filter of a system report depending on the value of the **Localize** property in the System Filter Columns section.

The System Filter Columns section of a system report includes a **Localize** property flag for each system filter. An application builder can use the internal value of the localizable field or the runtime value. By default, this is cleared for new and existing system reports. If an application builder checks this property, the query engine at runtime treats the text literal in the language of the user who is running the report. If it is cleared, the query engine filters the results based on the internal values from the main tables.

Language

The Language column in the community page shows the language in which the report was created.

Chapter 4. Exporting data and labels for translation

You can export labels and record data as XML Localization Interchange File Format (XLIFF) files. You can then translate the labels and data, and reimport the XLIFF files.

Exporting labels and record data for translation

If IBM TRIRIGA does not include your required translations, you can create customized translations by using the Globalization Manager to export labels and record data. This method is the preferred method for translating additions that you make. The exported language packs can then be translated by you or a third party, and the customized files can be imported into IBM TRIRIGA.

Before you begin

If you choose to do a delta export, the `tririga_version_date_language_packs.zip` file must be in the `<install directory>\userfiles\LanguagePacks` folder on the IBM TRIRIGA Application Platform when you do the export. The file is on the Application Upgrade image in the IBM TRIRIGA application eAssembly on the IBM® Passport Advantage® website.

Procedure

1. Select **Tools > Administration > Globalization Manager** and click **Export**.
2. Optional: In the **Product version info** field, enter information about the IBM TRIRIGA version. This information appears in the product-version attribute in the exported XLIFF files.
3. To export labels, ensure that the **Label** check box is selected and choose the scope of the labels to include.
4. To export data, ensure that the **Data** check box is selected and choose the scope of the business objects to include.
5. Select whether to do a full export or a delta export. A delta export affects only the labels and data that changed since the last export. If you select a delta export, you must also select the location of the reference files to which you want to compare the differences.
6. Select the target language for the export. Select **Internal Values** to export labels and data from the main tables that are included with the product. You can change the language values in the List Manager.
7. Click **Export**.

Export of database table indexes

In terms of globalization, the records that are created from certain Database Table Manager BOs, that is, `triIndex`, `triSparseColumn`, `triTable`, and `triIndexedColumn`, should not 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 BOs are always excluded during data export.

Storage of translation files

When you export record data and labels, the exported translation files are stored in the `userfiles\Translations` directory.

Record data

Exported record data is stored in the `userfiles\Translations\tririgaData\export` directory, with the following subdirectories:

`\updates`

Contains exported record data files. Files are organized by business object and named *Module_BusinessObject.xliff*. For example, the calendar period business object, which is in the Classification module, is exported to *Classification_triCalendarPeriod.xliff*.

Each XLIFF file for data contains a maximum of 100 records. If more than 100 records exist for a business object, extra files for that business object are created and numbers are appended to the file name, for example *Classification_triCalendarPeriod_01.xliff*, *Classification_triCalendarPeriod_02.xliff*.

Before each export, if the updates folder is not empty, the contents are backed up into the `\updates_date_time` folder.

`\reference`

Used as the base files for delta exports and updated after data files are exported. You can manually update the reference directory if required. Before each export, if the reference folder is not empty, the contents are backed up into the `\reference_date_time` folder.

`\forDeletion`

Contains files from the reference directory that are going to be deleted. Before each export, if the forDeletion folder is not empty, the contents are backed up into the `\forDeletion_date_time` folder.

`\deltaOnly`

Contains new and updated data from delta exports. Before each export, if the deltaOnly folder is not empty, the contents are backed up into the `\deltaOnly_date_time` folder.

Labels

Exported labels are stored in the `userfiles\Translations\tririgaLabel\export` directory, with the following subdirectories:

`\updates`

Contains exported label files. Files are organized by label type and named *Module_LabelType.xliff*. For example, the GUI labels, which are in the Document module, are exported to *Document_GUILabel.xliff*.

Each XLIFF file for labels contains a maximum of 1000 labels. If more than 1000 labels exist for a label type, extra files for that label type are created and numbers are appended to the file name, for example *Document_GUILabel_01.xliff*, *Document_GUILabel_02.xliff*.

Before each export, if the updates folder is not empty, the contents are backed up into the `\updates_date_time` folder.

\reference

Used as the base files for delta exports and updated after label files are exported. You can manually update the reference directory if required. Before each export, if the reference folder is not empty, the contents are backed up into the `\reference_date_time` folder.

\forDeletion

Contains files from the reference directory that are going to be deleted. Before each export, if the `forDeletion` folder is not empty, the contents are backed up into the `\forDeletion_date_time` folder.

\config

Contains the inclusion and exclusion files for the labels.

\deltaOnly

Contains new and updated labels from label delta exports. Before each export, if the `deltaOnly` folder is not empty, the contents are backed up into the `\deltaOnly_date_time` folder.

Chapter 5. Translating labels directly

As part of your globalization efforts, you must translate your labels into the base language and secondary languages. If you do not have a label language pack, you can use the Globalization Manager to translate the values directly.

About this task

You can translate values into any language that is defined in the List Manager. If the language you need is not already in the List Manager, you must add it.

The Globalization Manager does not contain a complete list of all fields that are in the application. You must export or import the files to obtain the entire list.

Procedure

1. Select **Tools > Administration > Globalization Manager**.
2. In the left pane, select the object that contains the field, message, or tooltip.
3. From the **Managed By** menu, select the type of field, message, or tooltip that you want to manage.
4. In the right pane, select the language that you want to translate into.
5. Enter translated values for the labels.
6. Save your changes.
7. Repeat for any additional pages of label values.

Chapter 6. Globalizing lists

Your applications use lists, such as radio button lists or drop-down lists, to manage and display information. With the List Manager, you can ensure that your users can interact with this information in their own languages.

Types of lists

IBM TRIRIGA provides lists of predefined values for many fields in the user interface. These lists contain valid data and you can select the required value from a list. You can improve data integrity by restricting users to select approved values from a list for entry into a field. Lists ensure that information is relevant, consistent, and correctly entered.

You can use the List Manager to create and manage radio button lists, drop-down lists of static values, dynamic lists derived from values in the database, and dependent lists.

In the List Manager, you can organize lists by the menu with which they are associated. For example, the Project menu has lists like project type, project business object type, project transaction type, severity, and work order type.

Static List

A Static list contains values that are fixed and do not change often, for example, states, colors, countries.

Dependent List

Dependent lists consist of two static lists, one parent list and one or more child lists. For example, a business object contains two list fields named Country and State. If the State list is defined as a dependent list, the list of values available for the State field is dependent on the Country selected.

Dynamic List

Dynamic lists are directly tied to a business object. For example, a new model number in a business object in the Asset menu. As new models are received, they are entered into the Asset business object. These model numbers are immediately available in a Model list that is used with a Parts Request form. You do not need to manually specify the new model numbers in the Parts Request form because the Model list was dynamically created from the Asset business object.

Specifying list items in alternative languages

You can modify the choices that appear in List fields, so users can see list choices in their preferred language.

About this task

Some lists do not need to be translated, for example, lists of postal abbreviations.

Procedure

1. Select **Tools > Administration > Lists**.
2. In the Manage By section, from the menu, select **Name**.

3. Select the list name for which you want to provide entries in another language.
4. In the right pane, click **Label List**, and select the language that you want to translate into.
5. Enter translated values for the list items.
6. Save your changes.

Chapter 7. Globalizing date and time

In IBM TRIRIGA, dates and times are defined in lists. With the List Manager, you can ensure that dates and times are displayed in the appropriate format for multicultural users.

Custom date and time formats

You can define different date formats or date time formats from the formats that are included with the IBM TRIRIGA Application Platform. In the List Manager, you can use the DateFormat list to control the formatting of Date fields and the DateTimeFormat list to control the formatting of Date and Time fields.

Table 5. Translatable date-only formats

| Format | Example |
|----------------|----------------|
| dd/MM/yyyy | 06/03/2007 |
| dd-MMM-yyyy | 06-Mar-2007 |
| MM/dd/yyyy | 03/06/2007 |
| MMM dd, yyyy | Mar 06, 2007 |
| MMMMM dd, yyyy | March 06, 2007 |
| yyyy.MM.dd | 2007.06.03 |
| yyyy/MM/dd | 2007/06/03 |
| yyyy-MM-dd | 2007-06-03 |

Table 6. Translatable date and time formats

| Format | Example |
|----------------------------|-----------------------------|
| dd-MM-yyyy HH'h'mm | 06-03-2007 13h44 |
| dd-MM-yyyy HH'h'MM'min' | 06-03-2007 13h44min |
| dd-MMM-yyyy HH:mm | 06-Mar-2007 13:44 |
| dd/MM/yyyy HH'h'mm | 06/03/2007 13h44 |
| dd/MM/yyyy HH:mm | 06/03/2007 13:44 |
| dd/MM/yyyy hh:mm a zzz | 06/03/2007 01:44 PM PST |
| dd/MM/yyyy HH:mm zzz | 06/03/2007 13:44 PST |
| dd/MM/yyyy HH:mm:ss | 06/03/2007 13:44:25 |
| dd/MM/yyyy hh:mm:ss a zzz | 06/03/2007 01:44:25 PM PST |
| dd.MM.yyyy HH:mm:ss | 06.03.2007 13:44:25 |
| MM/dd/yyyy HH:mm | 03/06/2007 13:44 |
| MM/dd/yyyy hh:mm a zzz | 03/06/2007 01:44 PM PST |
| MM/dd/yyyy HH:mm zzz | 03/06/2007 13:44 PST |
| MM/dd/yyyy HH:mm:ss | 03/06/2007 13:44:25 |
| MM/dd/yyyy hh:mm:ss a zzz | 03/06/2007 01:44:25 PM PST |
| MMMMM dd, yyyy hh:mm a zzz | March 06, 2007 01:44 PM PST |
| yyyy-MM-dd HH.mm | 2007-03-06 13.44 |

Table 6. Translatable date and time formats (continued)

| Format | Example |
|------------------------|-------------------------|
| yyyy-MM-dd hh:mm a zzz | 2007-03-06 01:44 PM PST |
| yyyy.MM.dd hh:mm a zzz | 2007.03.06 01:44 PM PST |
| yyyy/MM/dd hh:mm a zzz | 2007/03/06 01:44 PM PST |
| yyyy/MM/dd HH:mm zzz | 2007/03/06 13:44 PST |

The IBM TRIRIGA Application Platform does not directly control the language in which these names and abbreviations appear. They appear in the same language, independent of the language specified by users in their My Profile pages. The language in which the names and abbreviations appear is controlled by a layer of software that runs on top of the IBM TRIRIGA Application Platform. The specifics vary with the computing environment in which the platform runs.

Defining Date or Date and Time Format

If the format you want to use does not exist, you can define a different date format or date and time format.

About this task

Use the List Manager to complete this task.

Procedure

1. Select **Tools > Administration > Lists**.
2. In the Manage By section, expand the menu and select **Name**.
3. Select **DateFormat** or **DateTimeFormat**.
4. In the right panel, specify the new format in the **Value** field.
5. Save your entries.

Chapter 8. Globalizing currency

If your organization operates in multiple countries, you might need to manage different currency exchange rates.

Currency

Currencies play a vital role in financial transactions, especially for an organization that manages projects in multiple countries. In IBM TRIRIGA, you can establish a base currency in the TRIRIGAWEB.properties file, and you can automatically calculate values for other currencies.

Currency types, such as US Dollars and Euro, are in the Unit of Measure (UOM) values. You can find these currency types by selecting **Tools > Administration > Unit of Measure (UOM) > Values**. Limited support is available for globalization of currency types to your preferred language. When you create a record, and your My Profile record specifies a preferred currency, that currency is the default currency for all number fields that have Currency for their UOM. If you do not specify a currency in your My Profile record, the default currency is the default that is specified in the individual field.

If you have specified a currency default value, the currency fields display the currency's default display mask but with decimal and grouping separators that are based on your locale as specified by the **User Language** that is selected in your user profile.

When entering currency, if a grouping separator is specified, it must use the grouping separator based on the user's locale. If you enter a decimal separator, the decimal separator must be based on the user's locale. For example, if French is the user's language with a space as the grouping separator and a comma as the decimal separator to create a contract with US dollars as the currency UOM, the user may enter the value 12345.00 to display as \$12 345,00 in any of the following formats:

- 12 345,00
- 12345,00
- 12345

The Currency Conversion component is used with financial transactions. To access currency conversion rates, select **Tools > Administration > Currency Conversions**. You can find the current exchange rates and exchange rate history in the conversion groups. You can use the conversion groups to gather exchange rates from multiple sources. If a currency conversion rate does not exist for the posting date and time of a financial transaction, the most recent currency conversion rate is used.

You must evaluate your use of currencies before you add data records. If you add or remove currencies after you add records, the records might lose data or have other conversion issues.

A currency rate cannot be deleted if its start date and time value are in the past. The currency rate record is retained for historical purposes. If you click **Delete** on a currency rate whose start date and time value is in the future, the currency rate is deleted immediately.

Creating currency conversion rates

If you do not find the conversion rate that you want to use, you can create a conversion rate.

Procedure

1. Select **Tools > Administration > Currency Conversions**, and click **Add**.
2. Select the conversion group, from currency, and to currency. If the value you want to use for the fields is not available, you can use the List Manager to add the value.
3. Specify the conversion rate. You can use up to six decimal points for the conversion rate.
4. Specify the start date and time, and the end date and time.
5. Click **OK**.

Chapter 9. Globalizing units of measure

Every numerical data element (field) must have an associated Unit of Measure (UOM). For example, the unit of measure type “Area” can have “square-meters” as one of its UOM values.

You also can modify UOMs at the display properties field level by using the Form Builder tool.

When a numeric field defined in the Data Modeler points to a source field, the UOM values are read directly from that source field. A copy of the UOM information is not stored with the numeric field. UOM values for base currency fields are read directly from the TRIRIGAWEB.properties file. When you create a numeric field, the UOM parameters must be set explicitly. Unless specified, a UOM is not set for a newly created numeric field.

Evaluate your use of UOMs during implementation and add or remove UOMs pertinent to your company needs before you add data records. Failure to do so before you create records might cause conversion issues or data loss on those records.

Rounding rules for UOM values

When you configure unit of measure (UOM) values, the rounding rule defines how numbers are rounded to the number of decimal places specified in Storage Precision.

| Option | Description |
|-----------------|--|
| ROUND_CEILING | The fractional digit moves toward positive infinity. For example, if Storage Precision is 2, 1.234 is rounded to 1.24. |
| ROUND_DOWN | The fractional digit moves toward zero. The operation effectively truncates the value at the number of decimal places in Storage Precision. For example, if Storage Precision is 2, 1.2345 is rounded to 1.23. |
| ROUND_FLOOR | The fractional digit moves toward negative infinity. For example, if Storage Precision is 2, 1.234 is rounded to 1.23. |
| ROUND_HALF_DOWN | The fractional digit uses the digit to the right to determine the direction of the round. If the digit to the right is 6 or greater, the fraction digit is rounded up. If the digit to the right is 5 or less, the fractional digit is rounded down. For example, if Storage Precision is 2, 1.234 is rounded to 1.23, and 1.236 is rounded to 1.24. |
| ROUND_HALF_EVEN | The fractional digit is rounded up or down as needed to make it even. For example, if Storage Precision is 2, 1.234 is rounded to 1.24. |

| Option | Description |
|---------------|--|
| ROUND_HALF_UP | The fractional digit uses the digit to the right to determine the direction of the round. If the digit to the right is 5 or greater, the fractional digit is rounded up. If the digit to the right is 4 or less, the fractional digit is rounded down. For example, if Storage Precision is 2, 1.234 is rounded to 1.23, and 1.236 is rounded to 1.24. |
| ROUND_UP | The fractional digit moves away from zero. For example, if Storage Precision is 2, 1.234 is rounded to 1.24. |

Creating UOM types

The IBM TRIRIGA Application Builder has many standard UOM types, such as inch and volt. You can create additional UOM types to suit your requirements.

Procedure

1. Select **Tools > Administration > Unit of Measure (UOM) > Types** and click **Add**.
2. On the **General** tab, specify the name in the **UOM Type** field.
3. Optional: Specify a base UOM.
4. Click **Create**.
5. Click **Save & Close**.

What to do next

You can modify or delete the UOM type, but you must first evaluate your existing data to avoid conversion issues or data loss.

Creating UOM values

You can configure unit of measure (UOM) values such as the conversion factor, conversion offset, storage precision, and rounding rule.

Procedure

1. Select **Tools > Administration > Unit of Measure (UOM) > Values** and click **Add**.
2. In the **General** section, specify the UOM type.
3. In the **UOM Details** section, specify the relevant details.

Restriction: The grouping separator that is used in the **Display Mask** field must be the same as the grouping separator that is used in the **UOM Delimiter** value and the **UOM Decimal** entry. For example, a French currency UOM uses **### ###,00** as the display mask, a space as the UOM delimiter, and a comma as the UOM decimal. A UOM for the Brazilian Real that uses a period as the grouping separator and a comma as the decimal separator must use **###.###,00** as the display mask, a period as the UOM delimiter, and a comma as the UOM decimal.

4. Click **Create**.
5. Click **Save & Close**.

Chapter 10. Globalizing BIRT reports

The starter report and the IBM TRIRIGA BIRT integration have features to assist with the globalization of reports. Before you export labels for BIRT reports, make sure that these reports are set up in IBM TRIRIGA.

Dates

The IBM TRIRIGA BIRT integration uses IBM TRIRIGA as much as possible to localize dates. The display string for any IBM TRIRIGA Date or DateTime field is automatically formatted in the IBM TRIRIGA user's preferred date format. In addition, custom functions `formatDate()` and `formatDateTime()` allow you to present a BIRT Date or DateTime in the IBM TRIRIGA user's preferred date format.

Translations

IBM TRIRIGA supports the translation of all labels in the BIRT report. The starter report includes all the labels needed to translate the business object field labels in a report.

Field labels

A starter IBM TRIRIGA BIRT report is associated to a resource properties file with an extension of `tririgalabels.properties`. You can view this association in the report's Resources section in the **Property Editor – Report** tab.

This report resource points to a properties file that contains label keys for every business object field that was built into the report. The field label key uses the following format:

```
[FIELD]ModuleName__BusinessObjectName__FieldName
```

For example:

```
[FIELD]triPeople__triEmployee__triFirstNameTX  
[FIELD]triPeople__triEmployee__triLastNameTX
```

In the starter report, each key is already associated to its corresponding data set output column in BIRT.

If you use the Data Explorer as a canvas, you can drag elements onto the canvas without having to reassociate the display keys for those elements. If you do not drag elements onto the canvas, you can still use all the labels in the properties file on any report element that supports localization. You do this by selecting a report element in the layout and navigating to its **Localization** properties in the **Properties** tab of the **Property Editor – Label** tab. At runtime, text that uses these `[FIELD]` keys is translated based on the current user's language.

If you need a custom field label in your report that is not available from the starter report properties file, you can add keys and labels to the file to support those other scenarios. As a result, the custom labels are added to the `tririgalabels.properties` file. When the report design and the label files are exported and uploaded to IBM TRIRIGA, the custom labels from the BIRT report can be exported for translation from the Globalization Manager.

Non-field labels

The IBM TRIRIGA BIRT integration also supports labels such as parameter input labels and report titles.

The only requirement for a key name is that it cannot begin with the special [FIELD] prefix.

If you need a custom non-field label in your report that is not available from the starter report properties file, you can add keys and labels to the file to support those other scenarios. As a result, the custom labels are added to the `tririgalabels.properties` file. When the report design and the label files are exported and uploaded to IBM TRIRIGA, the custom labels from the BIRT report can be exported for translation from the Globalization Manager.

Libraries

A BIRT report library can house reusable report elements, including labels. You need to create a properties file with an extension of `tririgalabels.properties` and refer to that properties file as a library resource. To view a supported properties file for a BIRT report library, select the library in the BIRT Navigator, and view the properties in the **Property Editor – Library** tab.

To read the library, each BIRT report must reference the library before you export and upload the report to IBM TRIRIGA.

Chapter 11. Setting language preferences in profiles

When you sign in to the IBM TRIRIGA Application Platform, text and messages are presented in your preferred language. The default language is US English. When text is available in a secondary language, you can change your language in your user profile.

Procedure

1. Select **Portfolio > People > My Profile**.
2. On the **My Profile** tab, in the Preferences section, select the language and currency.
3. Optional: Select the time zone, date format, and area unit default. If you do not set these values, the default server values are used.
4. Optional: Enable bidirectional support and select the direction of user-generated text. If you select **Contextual**, the direction of text is set as described by Unicode. For more information, see http://www.unicode.org/reports/tr9/#The_Paragraph_Level. If you select **Default**, the direction of text is set to the default of the web browser.
5. Click **Save & Close**.

Chapter 12. Globalizing the Gantt scheduler

The Gantt scheduler is globalized for your multicultural users.

Changing the Gantt language

The language files for the Gantt scheduler are in several locations.

About this task

- Most of the Gantt tooltips, messages, and dialogs are in properties files that are in the `\Server Root/userfiles/Translations/Gantt/` folder on your IBM TRIRIGA application server. The file names are `TextXML_XX.properties`, where `XX` is the Java code for the language.
- Some of the Gantt words are in the as-delivered language packs. The `xliff` file names start with `SYSTEMLABEL_LEGACY`.
- The help text that is shown when the user clicks the help button in the Gantt section is in the `ibm-tririga.war` file. The files in the war file are named `Help_XX.html`, where `XX` is the language of the user. These files cannot be changed or added to.

Procedure

1. To change a translation for the Gantt tooltips, messages, and dialogs, edit the file `TextXML_XX.properties` in the `Server Root/userfiles/Translations/Gantt/` folder on your IBM TRIRIGA application server, where `XX` is the Java language code.
 - To add translations for a new language, copy the `TextXML.properties` file and update the name to be `TextXML_XX.properties`, where the `XX` is the Java language code.
 - If you are extending language support, do not use the language code plus the country code; the new file must have the language code. Within your new file, edit the values to the language that you are creating. Do not modify any comments or move around any entries. After the file is saved, test with a user of that language to make sure that no problems occurred with conversion of the file from `TextXML_XX.properties` into XML on the server.
2. To change a translation for the Gantt words in the as-delivered language packs, export them by using the System Labels option in the Globalization Manager export form. Look for `xliff` files that start with `SYSTEMLABEL_LEGACY`.

Chapter 13. Troubleshooting globalization

Review the troubleshooting tips when you encounter globalization issues.

Troubleshooting import of language packs

When you attempt to import data language packs, errors might occur. For example, the language pack might not import correctly, or the column size might be insufficient. You can view warnings and errors in the `TranslationData.log` and `LabelManager.log` files.

Import process is longer than expected

The import process for data normally takes less than 2 minutes. The import process for labels normally takes less than 7 minutes. However, the import process can take longer if the amount of data to be imported is large, if the system contains a large amount of data, or if you are using a customized environment. The increased length of import for data occurs because all path fields of all records in the target environment must be recalculated.

Insufficient column size error

In the Data Modeler, use the following steps:

1. Fix the metadata of the field by increasing its size.
2. Publish the business object.
3. Reimport the language pack or the problematic XLIFF file in the language pack.

The following table shows the error codes for each database that signify an insufficient column size.

Table 7. Error codes for insufficient column size

| Database | Error |
|----------------------|---|
| Oracle | ORA-12899: value too large for column |
| Microsoft SQL Server | Data truncation |
| IBM DB2 | DB2 SQL Error: SQLCODE=-302, SQLSTATE=22001 |

Language packs are not imported

Verify that the record ID or name of the translation data matches the record ID or name of the target environment.

This problem can occur if you delete or add as-delivered records in the target environment, and the row numbers of the smart section records are not the same as the row numbers in the language pack.

This problem can also occur if multiple languages are found for the language code that is specified in the target language of the import file. Check the logs, including the rolled over logs, for the error

There are multiple TRIRIGA languages available for the language: zh

If the error is in the logs, specify the correct language code by including the country code if needed in the language files and reimport. In the example error above, you need to add the country code that corresponds to your target language such as zh_CN or zh_TW.

This problem can also occur if the value in the target language in the import file is not in the Language Code table. Check the logs, including the rolled over logs, for the error

There is no available TRIRIGA language equivalent for the language: th

If the error is in the logs, add the target language to the Language Code table and reimport the files. For information on adding the target language, see *Creating language codes*.

Troubleshooting record data globalization

When you attempt to localize record data, errors might occur. For example, localized data might fail to display or might display in the incorrect language.

Localized data does not display in the record form or report

Use the following steps to determine why your data does not display:

1. Verify that the business object field is marked as localizable in the Data Modeler.
2. If the field cannot be marked as localizable in the Data Modeler, check the localizable property of its source field.
3. Check that the localized data is stored in the language table.
4. Check the `server.log` file and the `TranslationData.log` file for warnings and errors that relate to the field.

Empty values are displayed in the report for the localized field

Use the following steps to determine why empty values display in the report:

1. Verify that localized values exist in the language table (L_) for the user language.
2. If the field is a reference field, make sure that the localized value exists in the language table (L_) of the source field.

Empty values are displayed in module level reports or an error is displayed

Verify that the localized fields of the child business objects in the module are synchronized with the localized fields of the base business object. For example, if the `triNameTX` field is localized in the Classification base business object, then the `triNameTX` field of the `triAreaType` business object in the same module must also be localized.

Error during delta export

During a delta export, an error similar to the following error can occur.

An error occurred while processing the export request: [MID-2544015618]

The error can occur if the language packs .zip file is not in the `<install directory>\userfiles\LanguagePacks` folder on the IBM TRIRIGA server.

The error can also occur if the language packs .zip file is renamed. The language packs .zip file is a requirement for doing delta export versus language packs.

The IBM TRIRIGA installer no longer includes the `tririga_version_date_language_packs.zip` file. The file is on the Application Upgrade image in the IBM TRIRIGA application eAssembly on the IBM® Passport Advantage® website. The file must be downloaded and copied to the `<install directory>\userfiles\LanguagePacks` folder.

FileSystemAlreadyExistsException in the TranslationData.log file

This error can occur with delta exports. Delta export requests are asynchronous processes. If the second request is made before the first asynchronous process is completed, the first process still has a handle on the file. Wait for a little while until the first process completes before you start the second one.

Troubleshooting UX globalization

When you attempt to localize elements that support the UX framework, such as dictionary entries, errors might occur.

Translated static text does not display in the UX view

Translated UX File Cache

Clear the **Translated UX File Cache** in the Cache Manager. When you clear this cache, the components and HTML files that make up the translated UX views are refreshed. Translated UX views are regenerated the next time that they are requested from the server.

Translated Dictionary Cache

Clear the **Translated Dictionary Cache** in the Cache Manager. When you clear this cache, the translatable and translated text that is stored as records of the `triDictionaryEntry` business object in the System module are refreshed. Translated UX views are regenerated the next time that they are requested from the server.

If you change dictionary records from the Globalization Manager tool or forms, non-US English users must also clear their browser cache so that the translated view is regenerated with the updated translations.

T_TRIDITIONARYENTRY table

This issue can occur when you manually add a non-US English dictionary record for which there is no equivalent base language dictionary record defined. Before you manually add such a record, you must make sure that the `T_TRIDITIONARYENTRY` database table has an entry for the same text value, where the language code equals that of the base language.

The steps illustrate an example where the Danish translation for the phrase, Amount of paper waste is added.

1. Obtain the language ID values for both the base language and Danish. One method to obtain the language ID values is to run the following SQL query:

```
select * from list_value where list_id=975 and language_id=1;
```
2. Note the `LIST_VALUE_ID` for the base language and the language that is your target language.
3. Create a dictionary entry, where
 - `triLanguageIdNU` is the value for the base language (US English is 1)

- triBaseTX and triTranslationTX are the text in the base language (in this example, Amount of paper waste).
4. Create a dictionary entry for the Danish value, where
- triLanguageIdNU is the value for the target language that you obtained in step 2
 - triBaseTX is the text in the base language (in this example, Amount of paper waste)
 - triTranslationTX is the text in the target language (in this example, Mængde papiraffald)

Result: When you export the dictionary in Danish, the second record is displayed.

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 <http://www.ibm.com/privacy/details> in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy/>.



Printed in USA