TEAMCENTER

Teamcenter Localization

Teamcenter 2412

SIEMENS

About Siemens Digital Industries Software

Siemens Digital Industries Software is a global leader in the growing field of product lifecycle management (PLM), manufacturing operations management (MOM), and electronic design automation (EDA) software, hardware, and services. Siemens works with more than 100,000 customers, leading the digitalization of their planning and manufacturing processes. At Siemens Digital Industries Software, we blur the boundaries between industry domains by integrating the virtual and physical, hardware and software, design and manufacturing worlds. With the rapid pace of innovation, digitalization is no longer tomorrow’s idea. We take what the future promises tomorrow and make it real for our customers today. Where today meets tomorrow. Our culture encourages creativity, welcomes fresh thinking and focuses on growth, so our people, our business, and our customers can achieve their full potential.

Support Center: **support.sw.siemens.com**

Send Feedback on Documentation: **support.sw.siemens.com/doc_feedback_form**

Contents

Non-English locale environment settings

Special considerations when localizing Teamcenter applications

Troubleshooting

1. Overview of Teamcenter localization

Before you begin

Prerequisites	Depending on the type of localization you plan to do, Teamcenter administrator privileges may be required. • On Linux systems, the **LANG** and **LC_ALL** system environment variables must be set before you begin installing Teamcenter. Teamcenter Environment Manager (TEM) requires these settings. **LANG** and **LC_ALL** must be identical for installation tools to function properly. • Displaying localized versions of the standard Teamcenter user interface You do not need special Teamcenter privileges to deploy **localized versions of Teamcenter**. • **Localizing property values** in Teamcenter • Administrator rights Business Modeler IDE users must be members of the database administrators (**dba**) group on the Teamcenter server. Use the Organization application in the Teamcenter rich client to add a user to the **dba** group. • Importing localization files You must have **Translator** role privileges to **import translations**.
Configure Teamcenter	You must configure Teamcenter to enable localization. • Localized versions of the standard Teamcenter user interface **Certain Teamcenter preferences** are required to deploy localized versions of Teamcenter. • Localizing property values in Teamcenter You must set several Teamcenter **preferences to deploy localized property values** in Teamcenter.

What is Teamcenter localization?

Localization is the process of converting a program, such as Teamcenter, to run in a particular locale (a particular language spoken within a country) so that all text is displayed in the native language. The term *localization* refers to adapting software to a specific locale's language.

A *locale* represents a language and a country. For example, English is spoken in both the United States and England. However, there are differences in the English language between these countries. Therefore, there are different locale designators. Teamcenter supports English as spoken in the United States. When installing Teamcenter, it is automatically set to the **en_US** locale, where **en** is the English language identifier and **US** is the United States country identifier. The **fr_FR** locale is another supported Teamcenter locale, where **fr** is the French language identifier and **FR** is the France country identifier.

Display names are user-friendly aliases for internal object names.

Properties describe characteristics of a business object, such as name, serial number, and description.

There are several interface components that can be configured in local languages.

Teamcenter client user interface	When you purchase Teamcenter, you receive localization files that allow you to display the client interface in a number of standard languages (locales). To display the interface in these locales, you must install the localization files and configure the system environment and Teamcenter environment. Standard localization files do not have to be deployed, they must only be installed. You can modify the standard localization files to create your own localization of the Teamcenter interface. When you modify or create localization files, you must build and **deploy them**.
Property values (attributes)	Teamcenter can be configured to allow privileged users to specify localized values for business object properties. Some object property localization is available in the standard Teamcenter installation. For example, you can use the **Localization** button in Organization to add localized group names and in Query Builder to add localized query names and descriptions. You can configure most attributes and form fields to accept **localized values**.
Display names for data model components	You can modify the display name of standard and custom business objects, properties, and lists of values (LOVs) after they are created by using the **localization table in Business Modeler IDE**. This allows you to specify the name of the object in different languages or to simply change the name of the object in the default locale.

Supported Teamcenter localizations

Siemens Digital Industries Software provides localized versions of Teamcenter in the following locales:

Language	Locale code
Chinese (Simplified)	**zh_CN**
Chinese (Traditional)	**zh_TW**
Czech	**cs_CZ**
English	**en_US**
French	**fr_FR**
German	**de_DE**
Italian	**it_IT**
Japanese	**ja_JP**
Korean	**ko_KR**
Polish	**pl_PL**
Portuguese (Brazilian)	**pt_BR**
Russian	**ru_RU**
Spanish	**es_ES**

Use the appropriate locale codes to deploy Teamcenter localizations or launch Teamcenter clients in a desired locale.

If you provide your own localizations for locales not provided by Siemens Digital Industries Software, use the appropriate Java standard locale codes similar to the locale codes in the preceding table.[1]

Localizing Teamcenter in Hebrew

Siemens Digital Industries Software does not provide a Hebrew translation but provides recommended configuration settings for Hebrew locales. In Hebrew locales, set the locale code to **en_US**. This allows data entry in Hebrew, but user interface text is in English.

Localizing property values in Teamcenter

The **Localization** button allows localization administrators to enter localized text for property values. To place the **Localization** button on properties in Teamcenter, an administrator must use the Business Modeler IDE to set the **Localizable** property constant to **true** for those properties.

Using the **Localization** button, you can perform the following tasks:

- View the existing localization text.
- Modify the existing localization text.

1 Standard locale codes are composed of a two lowercase character language code from the ISO 639-1 standard, followed by an underscore, followed by a two uppercase character country code from the ISO 3166-1-alpha-2 standard.

- Add a translation value for a new locale without altering the master value of the property.
- Remove a translation value without altering the master value of the property.

> Note:
>
> The localized contents of a property cannot exceed the length defined for the given property in the database schema. That is, the length of the contents of a localized property has the same length restriction as the contents of the property *before* it was localized.

Viewing, editing, and adding translations for a property using the Localization button

The **Localization** button provides an easy way for users to view, edit, and add translations for a given localizable property. For example, in the Query Builder application, you can select a query regarding employee information.

> Note:
>
> - The **Localization** button appears by default in the following applications:
>   - Organization (site, group, and calendar)
>   - Query Builder
>   - Report Builder
>   - Classification Admin
>
>   In addition, the **Localization** button appears on localized property fields. You must use Business Modeler IDE to create a customized template to localize the individual property fields.
> - Only privileged users (with the **Translation** privilege set in Access Manager) can add and edit the translations.

The presence of the **Localization** button indicates the dialog box content can be localized. For example, if the master locale is set to English, you can add a translation for this employee information query. Using the **Localization** button, you can perform the following tasks:

- View the existing localization text.
- Modify the existing localization text.
- Add a translation value for a new locale without altering the master value of the property.

- Remove a translation value without altering the master value of the property.

To localize the name of the saved query, **Admin – Employee Information** in the example, click the **Localization** button to the right of the **Name** box. The **Language Translations** dialog box is displayed.

The master locale and master value are retained. Click the plus button (+) to add a localized value. Click the minus button (–) to remove a localized value. When adding a new value, you must add text in the following fields:

- **Master Locale**

  Displays the locale set for the site, for example, English.

- **Master Value**

  Displays the original text. This is the value that is stored when a new object is created. The content of this field can only be changed from the main dialog box.

- **Value**

  Specifies the property value content.

- **Status**

  Specifies the status of the localization. From the list, select one of the following:

  - **Approved**

    The text change is approved for use.

  - **Invalid**

    The text change is not valid.

  - **Pending**

    The text change is pending approval.

  - **Review**

    The text change is in review.

- **Locale**

  Specifies the locale. The list displays the locales that have been made available by the administrator, for example:

- **French**
- **German**
- **Italian**
- **Spanish**

2. Displaying and deploying localized versions of Teamcenter

Localization process overview

Teamcenter provides localization files that allow you to display client interfaces in a set of standard languages with regional conventions, or **locales**. To display Teamcenter in these locales, you must install the localization files and configure the system environment and the Teamcenter environment.

If you modify the standard localization files or create localization files for nonstandard locales, you must build and redeploy the client components to display the changes. In addition, you must perform the configuration tasks required to display a localized version of Teamcenter.

Configuring localization for Active Workspace

Localization configuration tasks

What is localization?

Localization is the presentation of an application's text in the local language. You can install Active Workspace to be displayed in many different languages.

Why configure localization?

After installing Active Workspace to run on a localized Teamcenter server, additional setup may be required to present text in the local language.

What can I configure?

You can configure the following aspects of localization:

- **Configure Active Workspace for additional locales**.
- **Configure locales for visualization servers**.

What do I need to do before configuring?

Before you can configure localization, you must ensure that the server-side of Teamcenter is configured for the locales you want.

Where can I find out more about localization?

See *Teamcenter Localization* in the Teamcenter help.

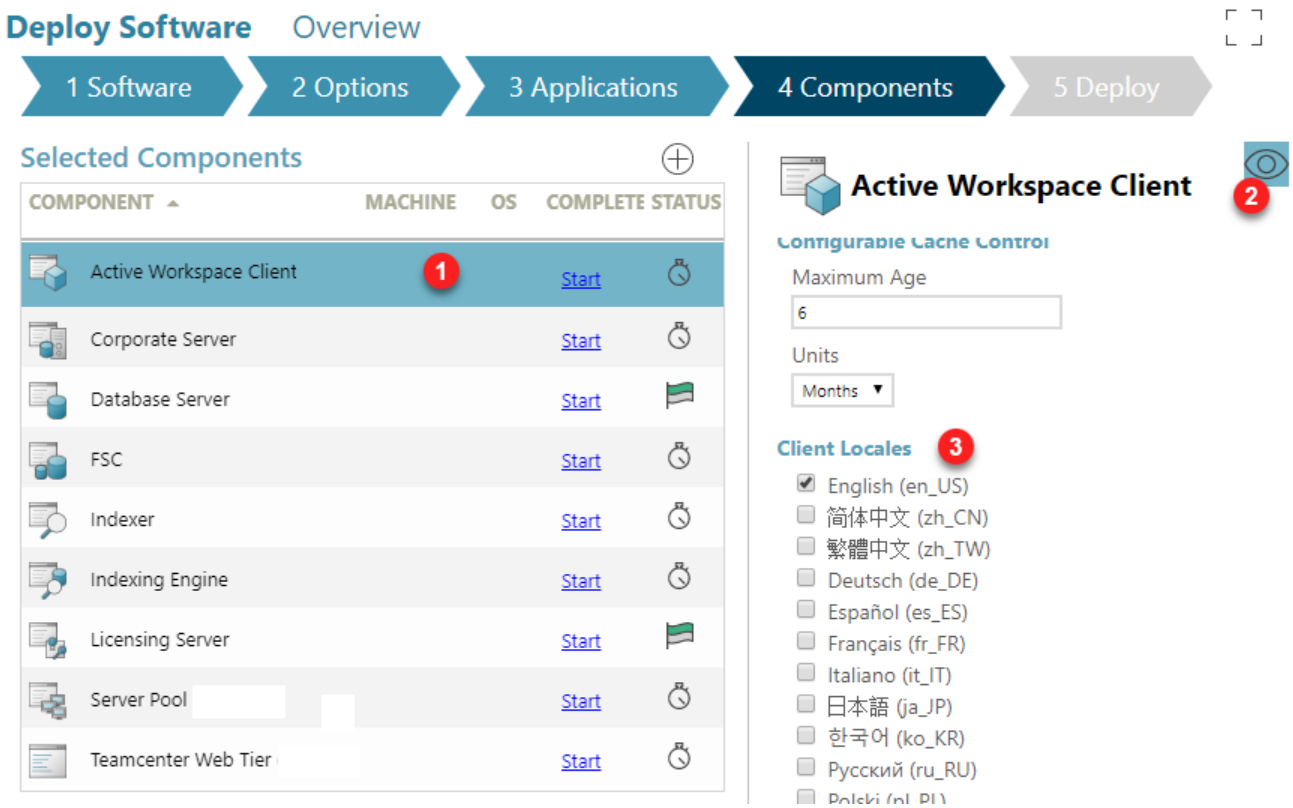
Configuring Active Workspace for other locales

You can configure Active Workspace to support various languages in addition to English. Use the Deployment Center or the Teamcenter Environment Manager to manage your available locales.

> Tip:
>
> When creating custom content, the parenthetical value is post-fixed to the JSON file name in the **i18n** folder of your custom module.

Deployment Center

The list of available locales is not a required parameter in the Deployment Center, so you must show all parameters when choosing locales.



Your list of components may differ.

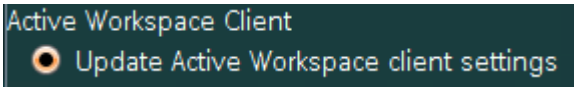
1. Select the **Active Workspace Client** component.

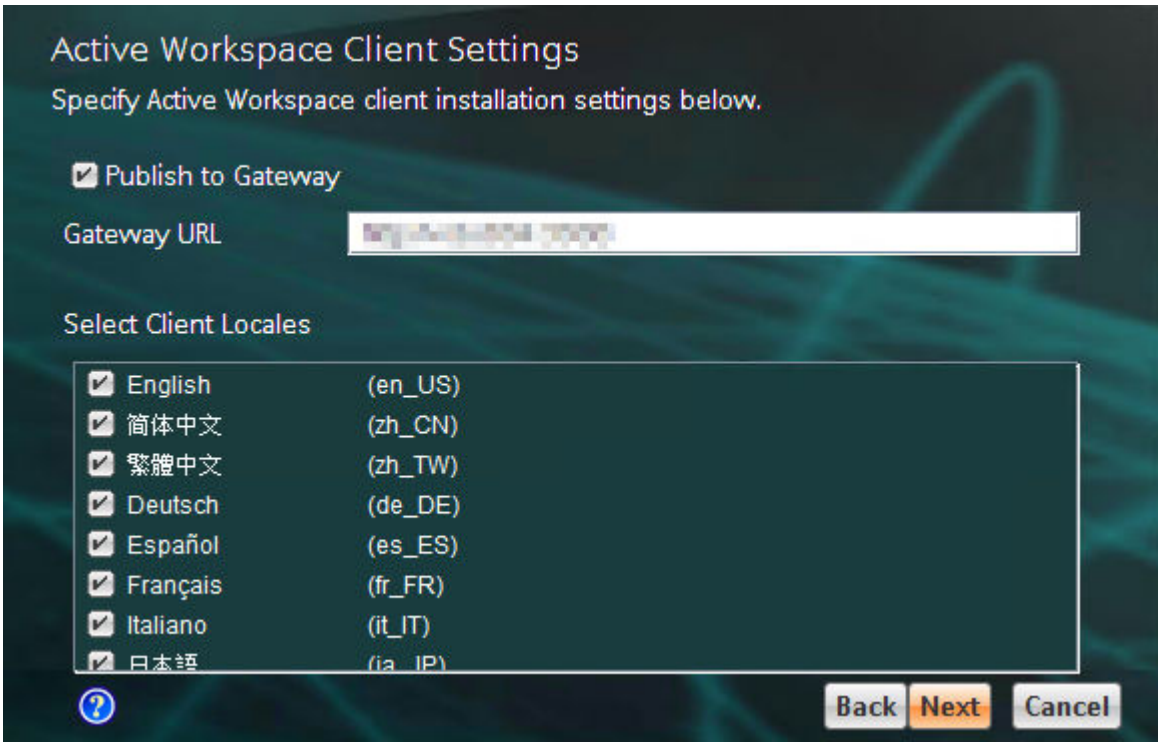
2. Select **Show all parameters**.

3. Choose your additional **Client Locales**.

Teamcenter Environment Manager

The list of available locales is found when you **Update Active Workspace client settings**.

1. 

2. 

Locale support by Visualization Server Manager

You can configure the Active Workspace client to display the user interface in any of the supported Teamcenter locales. However, some visualization data, such as Product and Manufacturing Information (PMI), requires a Visualization Server Manager (VSM) configured for the same locale as the information. For visualization data to display correctly in the Active Workspace client, you must have at least one VSM configured to run in each locale for which you have data. With this system in place, visualization processes are then routed to the appropriate server based on locale.

VSMs can be configured to support the following languages:

Brazilian Portuguese English Korean

Chinese (Simplified)	French	Polish
Chinese (Traditional)	German	Spanish
Czech	Italian	Russian
French	Japanese	

You can configure a VSM with any one of these languages. If you want to configure a cluster of VSMs to support more than one language, you need at least one VSM per language.

To change the language of a VSM, set Windows to the required language, location, and locale. You can adjust these settings using the Region and Language options found in the Windows Control Panel. You must adjust the **Date and time formats**, the **Current location**, and the **Current language for non-Unicode programs** values. After changing your Windows settings, reboot the system. When the VSM is started again, it inherits the new language configuration of the operating system.

If all VSMs are configured to use the same language, all clients use the available language regardless of browser preferences.

Note:

If you have a VSM system configured for two or more different languages, then Siemens Digital Industries Software highly recommends that at least one VSM be configured for English, even though this may require a minimum of three VSMs.

When the server system is configured with multiple languages, if at least one VSM is configured for English, then the English locale is a default.

The following table shows the VSM system response to a visualization data request from client when the client is not in one of the pre-configured languages.

VSM system configured for two or more languages	**Client is not in a pre-configured VSM language**
VSM for English exists	The data request is routed to an English VSM.
No VSM for English	The data request is rejected.

Configuring localization for the rich client

Configuring localized display of the Teamcenter client interface

Before users can choose the language (locale) in which the Teamcenter interface is displayed on their client machine, the locale must be installed on the server, and the following server-side administrative tasks must be performed:

- Define data entry locales.

  Set the **TC_language_data_entry** preference to define the locales for user data entry.

  > Note:
  >
  > The locales you specify for this preference must be supported by your database encoding.

- Specify the default language for the Teamcenter server process.

  Set the **TC_language_default** environment variable to set the default locale in the event that no locale is passed.

- Configure system environment settings.

  You must set your operating system and Teamcenter **environment settings for each locale** to ensure proper display and data processing.

Choose a display language

The default language displayed is the one specified by your operating system locale settings. You can choose to override the default display language if required.

At each logon, you can choose between multiple languages, depending on your company's policy and installation. Choose a language in the logon dialog when you log on to Teamcenter.

Alternatively, you can specify the language in your browser preferences. For example, in Microsoft Edge, choose **Settings→Languages** to add languages or modify language preferences.

Your ability to set the language for the client depends on the character set encoding of the Teamcenter server host and also the character set encoding of the Teamcenter database.

To prevent mixed-language display after you change the client display language, clear your web browser cache. This prevents the interface from displaying in mixed languages.

Choose a display language for the rich client

By default, the rich client is displayed in the language specified by the operating system. If you want to override the default language, you can choose the display language for the rich client.

Your ability to set the language for the rich client depends on the character set encoding of the Teamcenter server host and also the character set encoding of the Teamcenter database.

If you want to override the default language to launch the rich client in a desired language, add the **-nl** argument to the rich client launch command:

- Windows systems:

  TC_ROOT`\portal.bat -nl` *locale-code*

- Linux systems:

  TC_ROOT`/start_portal -nl` *locale-code*

Replace *TC_ROOT* with the Teamcenter home directory, and replace *locale-code* with the desired locale code .

For example, to launch the rich client Italian user interface, enter the following from a command prompt:

- Windows systems:

  ```
  D:\tc\rac\portal.bat -nl it_IT
  ```

- Linux systems:

  ```
  /tc/rac/start_portal -nl it_IT
  ```

Alternatively, on Windows systems, you can customize the properties for the Teamcenter rich client desktop shortcut icon  to specify a desired language:

1. On your desktop, right-click the Teamcenter rich client shortcut icon  .
2. Choose **Properties**.

   A properties dialog box is displayed.
3. Click the **Shortcut** tab.
4. In the **Target** box, add an **-nl** argument to specify the desired language.

The **-nl** argument accepts a single string as value. The string must be one of the **supported locale codes**.

For example, to run the rich client Italian user interface:

```
D:\tc\rac\portal.bat
```

becomes:

```
D:\tc\rac\portal.bat -nl it_IT
```

Note:

To prevent mixed-language display the next time you run the rich client after you change the **-nl** argument value, or after you change your operating system locale, delete the **Teamcenter** directory under your user directory.

- On Windows clients, delete:

  C:\Documents and Settings*user-name***Teamcenter**

- On Linux clients, delete:

  $HOME/Teamcenter[1]

If you find that Asian multibyte characters do not display correctly when you start the rich client, set your system font to a font that supports Asian multibyte characters. For example, on Windows systems other than Windows 10, the **Arial Unicode MS** font can be set to **Message Box** to correct this problem.

Similarly, if you find that Asian multibyte characters do not display correctly when you start the rich client using the native language (**-nl**) option, restart your system in the appropriate locale and set your system font to a font that supports Asian multibyte characters.

Caution:

If you use the Lifecycle Visualization embedded viewer, do *not* use the **-nl** argument when you launch the rich client.

For the embedded viewer to work properly, the operating system locale and the rich client runtime locale must match. The **-nl** argument overrides the Java locale and can cause incorrect behavior in the embedded viewer.

1 **$HOME** is your user home directory.

Example: Displaying the Teamcenter rich client in German

To change the Teamcenter display from English to German by using the **Properties** selection on your Teamcenter desktop shortcut, perform the following steps:

1. On your desktop, right-click the Teamcenter rich client shortcut icon  .

2. Choose **Properties**.

   A properties dialog box is displayed.

3. Click the **Shortcut** tab.

4. In the **Target** box, add an **-nl** argument to specify the desired language.

   The **-nl** argument accepts **a single string** as the value.

   For example, to run the rich client German user interface:

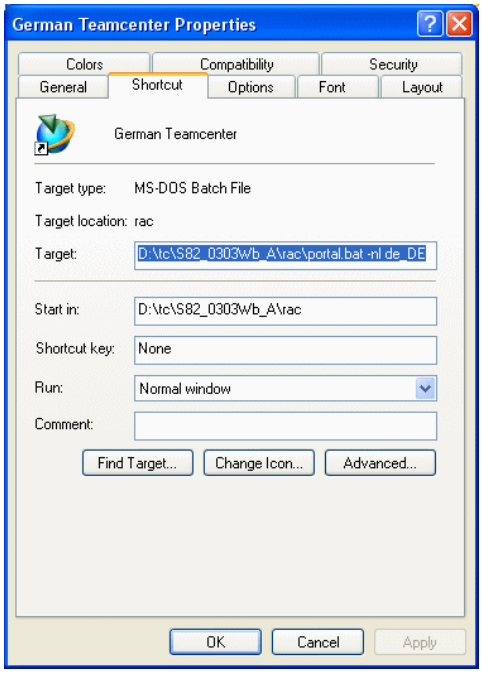
   ```
   D:\tc\rac\portal.bat
   ```

   becomes:

   ```
   D:\tc\rac\portal.bat -nl de_DE
   ```

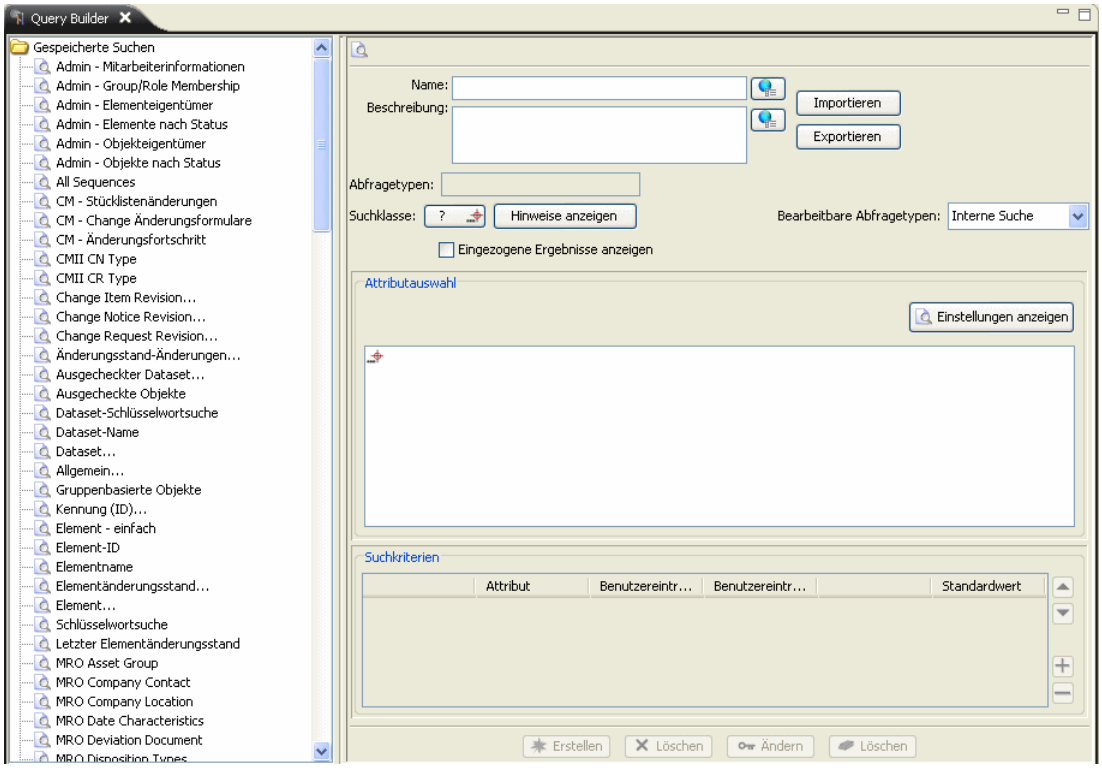
   > Caution:
   >
   > If you use the Lifecycle Visualization embedded viewer, do *not* use the **-nl** argument when you launch the rich client.
   >
   > For the embedded viewer to work properly, the operating system locale and the rich client runtime locale must match. The **-nl** argument overrides the Java locale and can cause incorrect behavior in the embedded viewer.

   The **German Teamcenter Properties** dialog box appears.



5. Click **OK**.

   Teamcenter displays in German.



Switching between display languages

When you install Teamcenter, the client interface displays in the language specified by your operating system locale settings. Teamcenter includes localization files for a number of locales. After the localization files are installed and the operating system and Teamcenter configuration tasks are complete, you can easily switch between languages on your client machine.

You can:

- Choose the display language in the rich client, by either:
  - Extending the command line.
  - Changing the **Properties** selection on your Teamcenter desktop shortcut (Windows systems).

Add multibyte character support in an English rich client

1. In the rich client **\rac\plugins\configuration_***config-name* directory, create the **customer.properties** file, if it does not already exist.

   Do not save the **customer.properties** file in Unicode or UTF-8 format. The **customer.properties** file must be in the default format (for example, ANSI) to be read successfully by the rich client.

2. Open the **customer.properties** in a plain text editor.

3. Add the following line to the file to set the **UseDefaultSwingFonts** property.

   ```
   UseDefaultSwingFonts=true
   ```

4. Save the file and exit the text editor.

5. Change to the **rac\registry** directory.

6. Run the **genregxml.bat** utility to register the change.

When you run Teamcenter in a multibyte environment, make sure the **TC_XML_ENCODING** environment variable is set to **UTF-8** and the **UGII_UTF8_MODE** environment variable is set to **1**.

Updating rich client localized text

If you change the strings of a localized rich client user interface, you must convert the files to Unicode and regenerate the Java archive (JAR) file.

To identify the localized user interface files you need, look for the language and country identifier added to the base resource file. For example, for the **aif_locale.properties** English resource file, which must remain in English, the equivalent Japanese file is named **aif_locale_ja_JP.properties**.

Convert native **.properties** files to Unicode as follows:

1. Copy the base file to a temporary file name, for example, from **aif_locale.properties** to **aif_locale_temp.properties**.

2. Edit the *base*_**locale**_*temp*.**properties** file, modifying the values to the correct native language.

3. Save the file.

4. Run the **native2ascii** utility against the temporary properties file to convert it to a Unicode properties file.

   The **native2ascii utility** is in the **\bin** directory of Java SDK 1.4.

   For example, to convert the properties file from Japanese to Unicode, enter the following command from the command line:

   ```
   native2ascii -encoding SJIS aif_locale_temp.properties aif_locale_ja_JP.properties
   ```

   The final locale-specific properties file or the output of the **native2ascii** file must have the *base*_**locale**_*locale-id_language-id*.**properties** file format.

   The rich client finds the value of a key in the following order:

   BASE_user.properties
   BASE_locale-ID_language-ID.properties
   BASE_locale.properties
   BASE.properties

5. Recompile the JAR file.

6. Install fonts if necessary.

   For information about fonts, see the Oracle Java web site.

For more information about converting files to Unicode, see the Unicode Consortium web site:

http://www.unicode.org/

3. Using localized property values in the client interface

Property value localization overview

Teamcenter provides features that allow privileged users to enter localized values on properties. Some of the Teamcenter applications, such as Organization, Query Builder, and Classification Admin include the capability to enter localized property values by default. For applications that do not include localized property value capability by default, you can use the Business Modeler IDE to enable localization on the properties of standard or custom business objects.

Caution:

The objects for which properties can be localized and the implementation procedures for localization can differ between Teamcenter applications. These differences affect the following applications:

- Audit Manager
- Classification
- Electronic Design Automation (EDA) considerations
- Manufacturing
- Mechatronics Process Management
- Localizing Schedule Manager
- Multi-Site Collaboration
- TC XML
- Structure Manager
- Workflow

How do you know if a property is localizable?

The **Localization** button  indicates the dialog box content can be localized and provides an easy way to view, edit, and add translations for a given localizable property.

Use the **Localization** button to:

- View the existing localization text.
- Modify the existing localization text.
- Add a translation value for a new locale without altering the master value of the property.
- Remove a translation value without altering the master value of the property.

Who can enter localized property values?

By default, the following privileged users can enter localized property values:

- **dba** users
- Owning users of objects
- Users in groups and roles that are granted access by the **Translation** access control list (ACL).

Enter localized property values

Only privileged users (with the **Translation privilege** set in Access Manager) can add and edit the translations.

1. Click the **Localization** button  next to the property field.
2. In the **Language Translations** dialog box, click the plus button (+) to add a localized value.

   > Note:
   >
   > The master locale and master value are retained when you add localized values to a property.

3. Enter text in the **Language Translations** dialog box fields:

Master Locale	Displays the locale set for the site.
Master Value	Displays the value specified in the master locale.
Value	Specifies the property value in a locale other than the master locale. This is where you enter the localized property value.
Status	Specifies the status of the localization. The following values are valid: • **Approved** The text change is approved for use. • **Invalid** The text change is not valid. • **Pending**

	The text change is pending approval. • **Review** The text change is in review.
Locale	Specifies the locale. The list displays the locales that have been made available by the administrator, for example: • **French** • **German** • **Italian** • **Spanish**

Searching for localized property values

Teamcenter performs two types of searches: locale search and scalar (nonlocale) search. Locale search returns localized property values; scalar search does not. The system determines which search to use based on whether the properties of the starting point of the search (type and class) are marked as being localizable. Localizable properties for a type/class are inherited by the child types/classes in the hierarchy. *However, locale search does not support hierarchical inheritance.* Therefore, to ensure that your search returns localized property values, you must initiate the search at a point in the hierarchy at which the properties are designated as being localizable.

Example:

Description is a property of the **Workspace Object** class. The **Description** property on the **Item_sub** item type (a subtype of **Item**) is marked as being localizable, but the parent class level (**Item**) is not.

A search is performed on the **Description** property of the **Item_sub** type, and because the property is localizable, the locale search is used and any localized descriptions are returned in the search results. However, when a search is performed on the **Description** property of the **Item** class, Teamcenter uses the scalar search because the **Item** properties are not marked as being localizable. The search does not return localized descriptions, even if the hierarchical search option is enabled.

Configuring property value localization

Create the Translation group and Translator role

To allow only a select group of users (who are neither **dba** users nor owning users) to modify translations for localizable properties, perform the following steps in Organization:

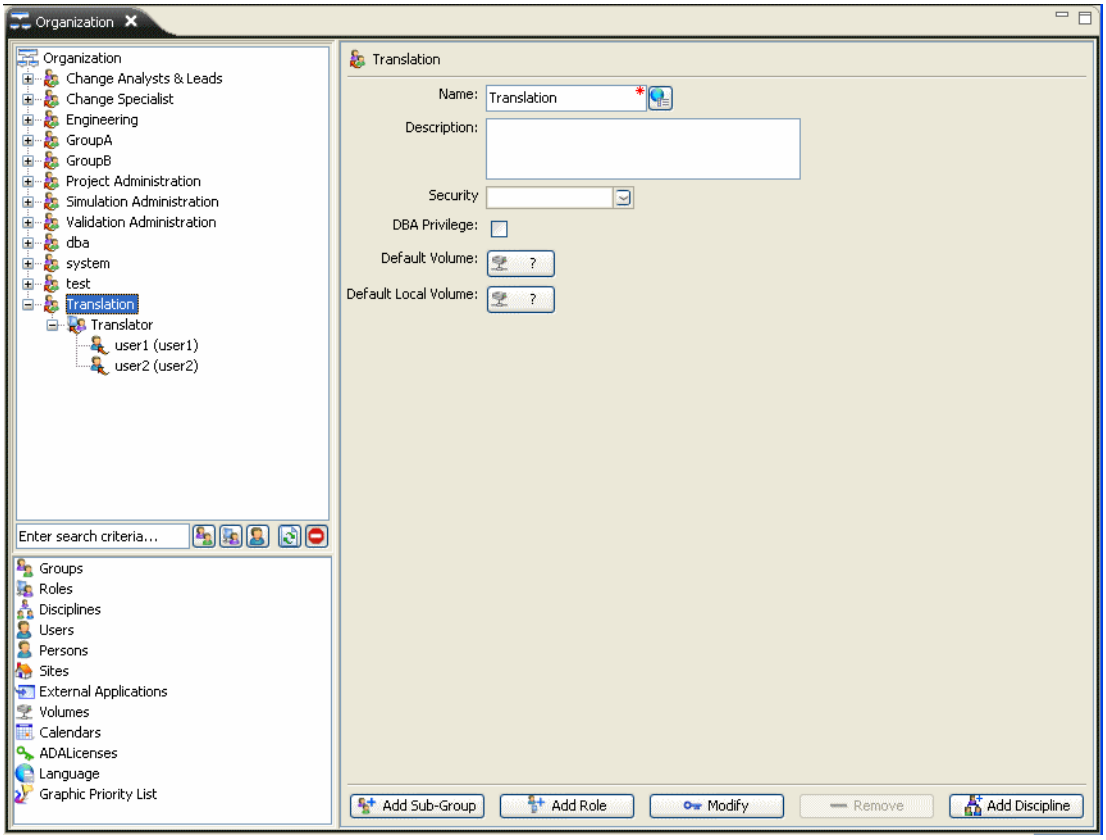
Note:

Creating a special group and role for translation privileges is optional. You can use an existing group and role to define the privileges required for translation access.

1. Create a new group, **Translation**, in the Organization application for translation. This group is where you add users who perform translation tasks.

2. Create the **Translator** role in Organization. Use this role for the users who perform translation tasks.

3. Add existing or new users to the **Translator** role.



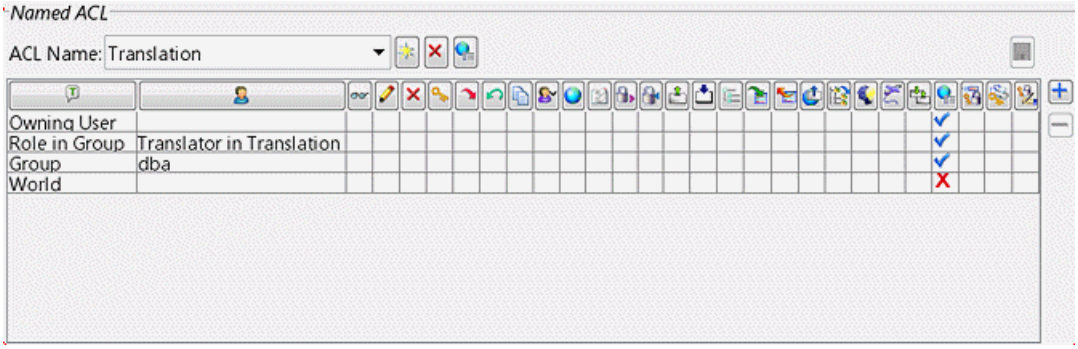
Configure the Translation access control list (ACL)

To allow select groups of users to view or modify translations for localizable properties using the **Localization** button, you must create a new rule in the **Translation** ACL.

Note:

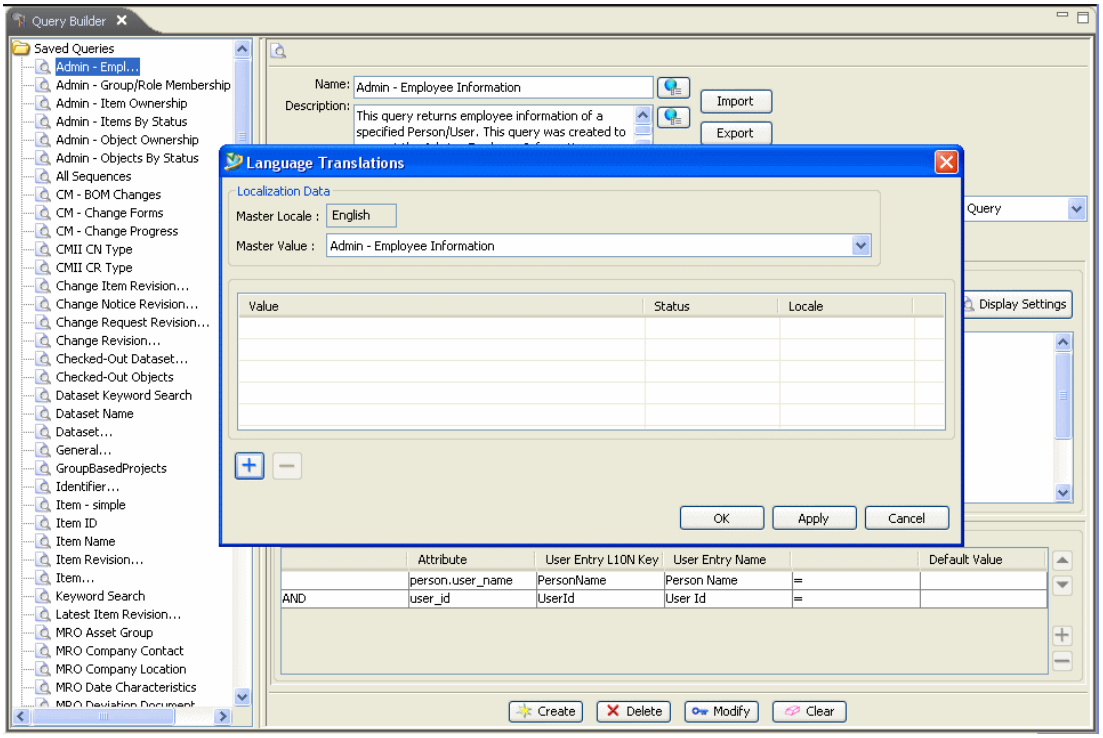
The new rule is in addition to the **dba** group, **Owning user**, and **World** entries in the ACL.

Using Access Manager, create a new **Role in Group** entry in the **Translation** ACL, and grant translation privilege to the role in group.



This grants users translation access privileges. When these users log on to Teamcenter, they see the **Localization** button in the properties dialog box next to a property object that is localizable.

The following figure shows the **Language Translations** dialog box that is displayed when you click the **Localization** button associated with the **Name** property.



Set preferences for localizing property values

Set the following preferences to localize property values in Teamcenter.

- **TC_language_localization_display**

To display translation information as read-only for properties that are localizable, set the value to **true**.

- **TC_language_localized_property_value_display**

  To specify the language to use for displaying localized property values, set the value to one of the following locale codes:

Locale code	Language
cs_CZ	Czech
de_DE	German
en_US	English
es_ES	Spanish
fr_FR	French
it_IT	Italian
ja_JP	Japanese
ko_KR	Korean
ru_RU	Russian
zh_CN	Chinese (as spoken in China)
zh_TW	Chinese (as spoken in Taiwan)

- **TC_language_localized_property_value_entry**

  To specify the data entry locale (master locale) of the logged-on user, set the value to one or more of the following locale codes:

  > Note:
  >
  > The master locale is the one used by the user to create/edit data in a localized property.

Locale code	Language
cs_CZ	Czech
de_DE	German
en_US	English
es_ES	Spanish
fr_FR	French

Locale code	Language
it_IT	Italian
ja_JP	Japanese
ko_KR	Korean
ru_RU	Russian
zh_CN	Chinese (as spoken in China)
zh_TW	Chinese (as spoken in Taiwan)

Limit languages for data entry

Set the **TC_language_data_entry** preference to limit the languages in which data can be entered in the **Language Translation** dialog box.

For example, to restrict data entry to English, French, German, and Spanish, set the following values for the **TC_language_data_entry** preference:

- **en_US**
- **fr_FR**
- **de_DE**
- **es_ES**

In this example, if you log on to Teamcenter using a language that is not specified for data entry, the system displays the following message, but does not prevent you from logging on.

```
Your data language must be done using the following locales: English,
French,
German, Spanish
```

4. Localizing operational data and data model elements

Operational data and data model element localization

Data model elements are text labels such as type names, relation names, and property names. *Operational data* are property values and lists of values. Both data model elements and operational data can be exported to XML localization files that can be viewed and edited using standard XML editing tools. The following are some of the tasks that can be performed using the localization files:

- Translate model elements and property values when adding a new locale to Teamcenter.
- Add translated names and values for objects that are not yet translated in a particular language.
- Translate data for objects created within a given date range.
- Translate data for all parts in an assembly.
- Review existing translations.

Operational data localization process

Property values can be translated using the **Localization** button  in the Teamcenter client interface or by using the following process:

- Determine which property values to localize.

  You can determine which property values to localize by running a query and then filtering the data based on specified criteria.

- Export the objects to an XML file using tools in the rich client or using the **l10n_import_export_utility**.
- Translate the contents of the XML file.
- Import the translated XML file into the Teamcenter database.
- Review or modify the translations by using the **Localization** button  corresponding to the property in the Teamcenter interface.

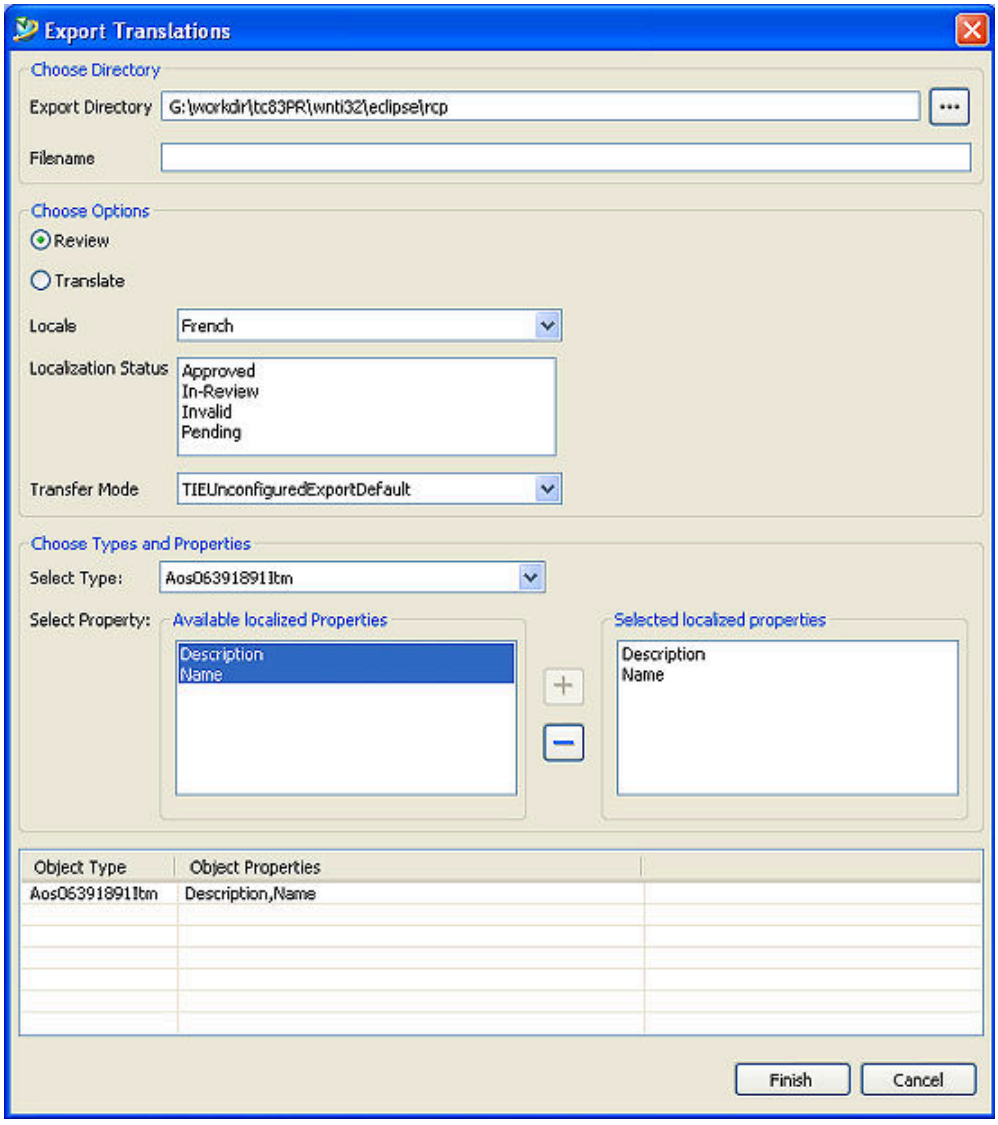
Note:

- You must have **Translation** privileges to import translations.
- You must use a UTF-8 compliant editor, such as Microsoft Word, to manipulate the translation files.

Export translations for review

You can export existing translations for a specific locale with specific statuses to an XML file that can be viewed and edited in an XML editor and imported back into Teamcenter.

1. With the objects you want to export selected, choose **Tools→Localization→Export Translations**.



2. In the **Export Translations** dialog box, select a directory for the XML export file to be saved in.

3. Type a name for the export file in the **Filename** box.

4. Select the **Review** option.

5. From the **Locale** list, select the locale for which you want to review translations.

6. From the **Localization Status** list, select the statuses of the translations that you want to view.

Approved	The text is approved for use.
In-Review	The text is in review.
Invalid	Text is considered invalid if the master language content is changed.
Pending	The text is pending approval.

You can select multiple status filters. Only localized properties that match the selected statuses are exported.

7. Select a transfer mode from the **Transfer Mode** list.

The transfer mode defines all related objects that must be candidates for translation of localizable properties.

8. In the **Choose Types and Properties** section, refine the list of object types and related properties for export by removing unwanted entries from the **Selected localized properties** list.

   a. Select an object type from the **Select Type** list.

      The **Available localized properties** and **Selected localized properties** lists are populated with the localized properties for the object type. These lists are derived from the objects initially selected for export. Localizable properties for related objects determined by the transfer mode are not included in these lists.

   b. Select the properties in the **Selected localized properties** list that you do not want to export and click —.

   c. Repeat the previous steps for each object type in the **Select Type** list.

   The selected object types and corresponding properties are displayed in the table at the bottom of the **Export Translations** dialog box.

9. Click **Finish**.

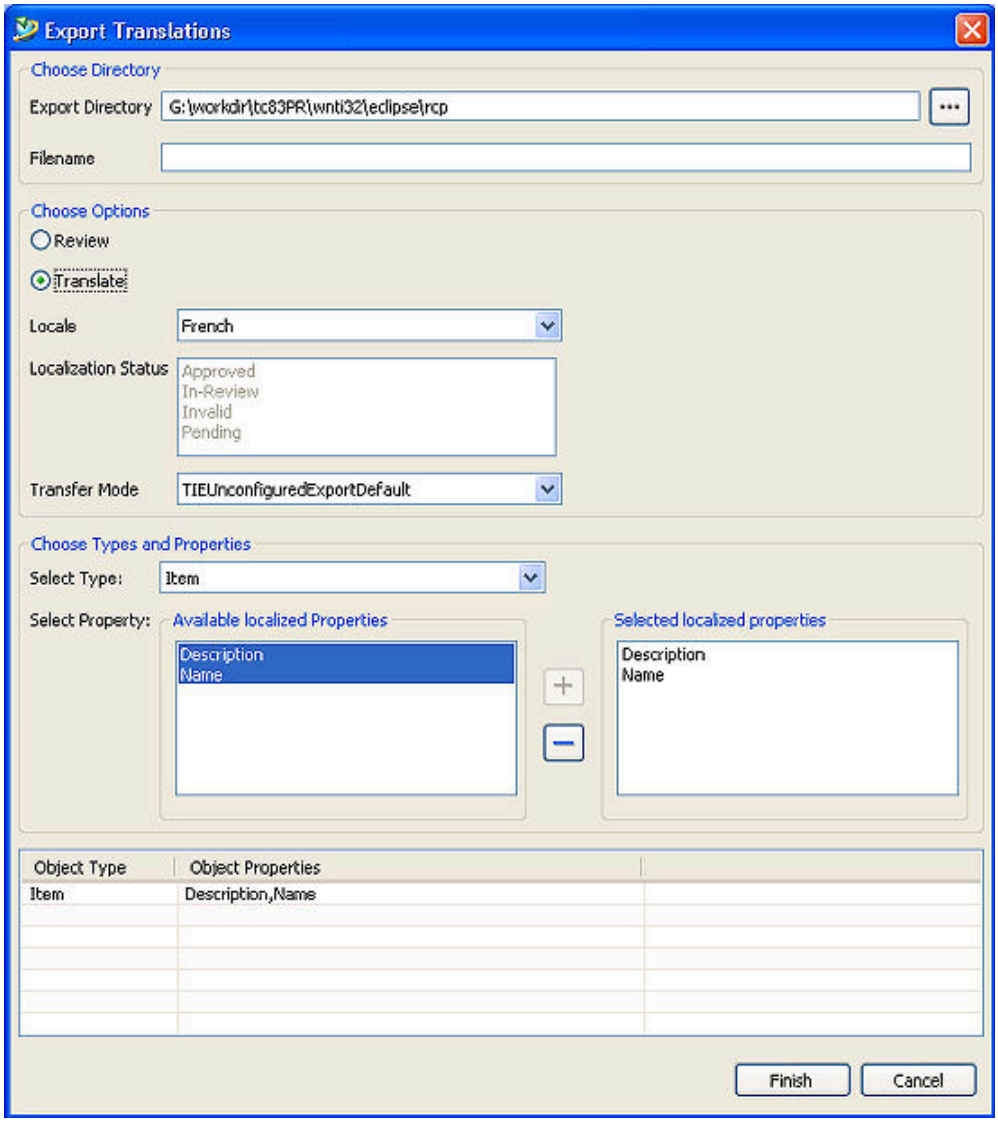
The system generates an XML file containing the property name, master value, translation value, and status of the exported localizable properties for each object. The file is located in the directory that you specified during the export process.

Export untranslated master values for translation

You can export master values for objects that do not have translations for a selected locale to an XML file that can be viewed and edited in an XML editor. You can also make copies of the XML file to use for generating translations in multiple locales.

> Note:
>
> Because each localizable property can have a master locale, it is possible that a set of objects could have multiple master locales. If multiple master locales exist for a set of objects, a separate XML file is generated for each master locale on the properties.

1. With the objects you want to export selected, choose **Tools→Localization→Export Translations**.



2. In the **Export Translations** dialog box, select a directory for the XML export file to be saved in.

3. Type a name for the export file in the **Filename** box.

4. Select the **Translate** option.

5. From the **Locale** list, select the locale for which you want to export localizable properties.

6. Select a transfer mode from the **Transfer Mode** list.

   The transfer mode defines all related objects that must be candidates for translation of localizable properties.

7. In the **Choose Types and Properties** section, refine the list of object types and related properties for export by removing unwanted entries from the **Selected localized properties** list.

a. Select an object type from the **Select Type** list.

   The **Available localized properties** and **Selected localized properties** lists are populated with the localized properties for the object type. These lists are derived from the objects initially selected for export. Localizable properties for related objects determined by the transfer mode are not included in these lists.

b. Select the properties in the **Selected localized properties** list that you do not want to export and click −.

c. Repeat the previous steps for each object type in the **Select Type** list.

The selected object types and corresponding properties are displayed in the table at the bottom of the **Export Translations** dialog box.

8. Click **Finish**.

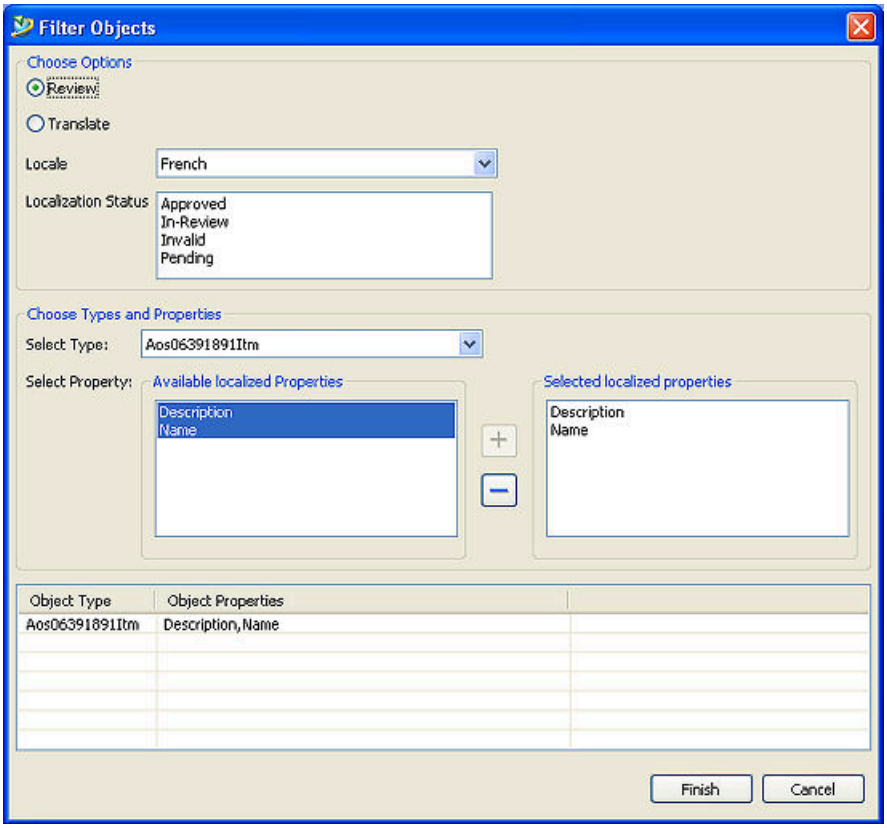
The system generates an XML file containing the property name, master value, translation value, and status of the exported localizable properties for each object. The file is located in the directory that you specified during the export process.

Review translations in the rich client

You can filter objects in the rich client to review existing translations or add translations interactively using the **Localization** button  .

The **Filter Objects** feature is similar to the **Export Translations** feature, but rather than generating an XML file, the output is displayed in the **L10N Results View**. In addition, the **Filter Objects** feature does not use transfer modes. Only the objects that match the selected locale and status criteria are displayed.

1. With the objects you want to filter selected, choose **Tools→Localization→Filter Objects**.



2. In the **Filter Objects** dialog box, select the **Review** option.

3. From the **Locale** list, select the locale for which you want to review translations.

4. From the **Localization Status** list, select the statuses of the translations that you want to view.

Approved	The text is approved for use.
In-Review	The text is in review.
Invalid	Text is considered invalid if the master language content is changed.
Pending	The text is pending approval.

You can select multiple status filters. Only localized properties that match the selected statuses are displayed.

5. In the **Choose Types and Properties** section, refine the list of object types and related properties for display by removing unwanted entries from the **Selected localized properties** list.

   a. Select an object type from the **Select Type** list.

      The **Available localized properties** and **Selected localized properties** lists are populated with the localized properties for the object type.

b. Select the properties in the **Selected localized properties** list that you do not want to display and click —.

c. Repeat the previous steps for each object type in the **Select Type** list.

The selected object types and corresponding properties are displayed in the table at the bottom of the **Filter Objects** dialog box.

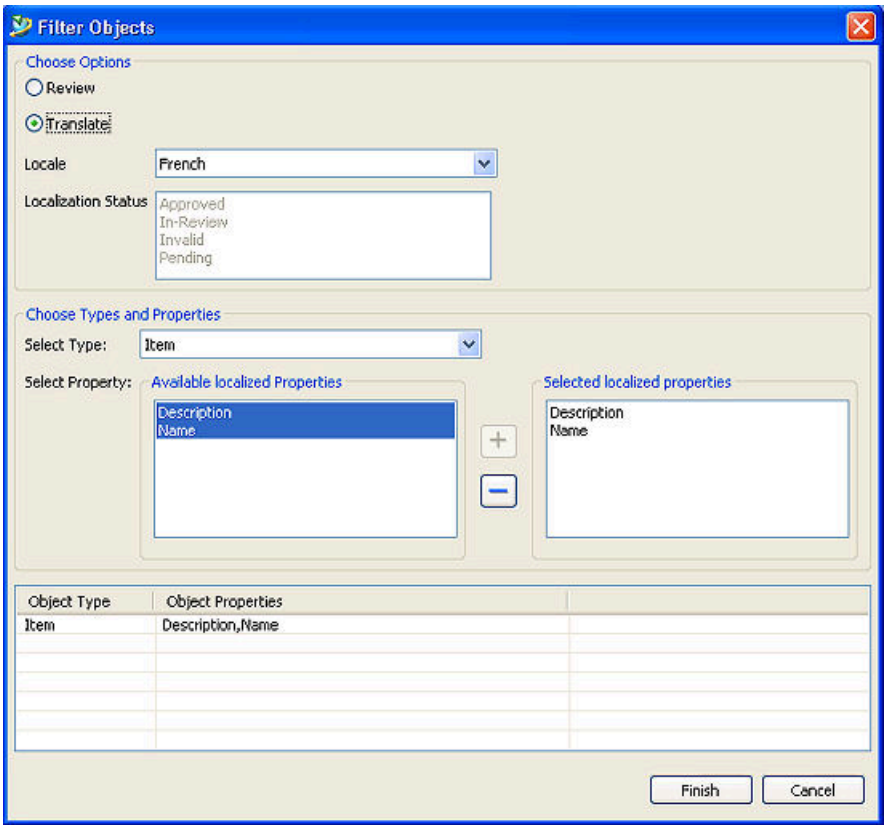
6. Click **Finish**.

The objects matching the filtering criteria are displayed in the **L10N Results View**.

Find untranslated master values in the rich client

You can filter master values for objects that do not have translations for a selected locale and display them in the **L10N Results View** in the rich client.

1. With the objects you want to filter selected, choose **Tools→Localization→Filter Objects**.



2. Select the **Translate** option.

3. From the **Locale** list, select the locale for which you want to display localizable properties.

4. In the **Choose Types and Properties** section, refine the list of object types and related properties for display by removing unwanted entries from the **Selected localized properties** list.

   a. Select an object type from the **Select Type** list.

      The **Available localized properties** and **Selected localized properties** lists are populated with the localized properties for the object type.

   b. Select the properties in the **Selected localized properties** list that you do not want to display and click —.

   c. Repeat the previous steps for each object type in the **Select Type** list.

   The selected object types and corresponding properties are displayed in the table at the bottom of the **Filter Objects** dialog box.

5. Click **Finish**.

The objects matching the filtering criteria are displayed in the **L10N Results View**.

Import localization files using the rich client interface

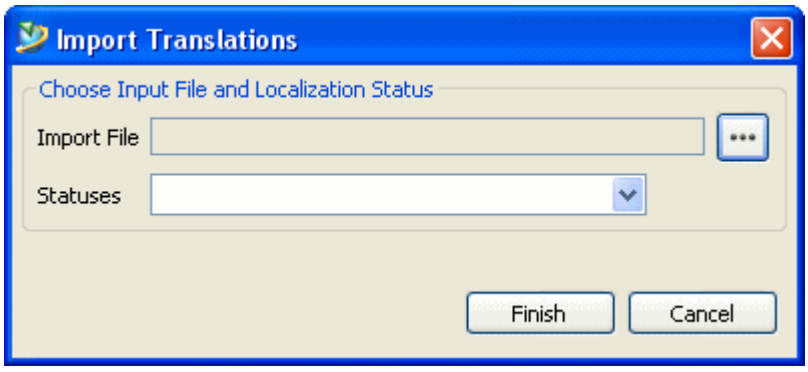
Use My Teamcenter in the rich client to import translated object properties from an XML file into the Teamcenter database.

You must have **Translation** privileges to import translations.

> Note:
>
> You can also import localization data from an XML file using the **l10n_import_export_utility** utility.

1. In My Teamcenter, choose **Tools→Localization→Import Translations**.

   The system displays the **Import Translations** dialog box.



2. Click the **Browse** ••• button to the right of the **Import File** box to select a valid XML translation file.

3. Select a localization status from the **Statuses** list.

   This status is applied to all translations imported from the file.

   - **Approved**

     The text is approved for use.

   - **In-Review**

     The text is in review.

   - **Invalid**

     The text is not valid.

   - **Pending**

     The text is pending approval.

4. Click **Finish** to import the translated content.

Localization using the Business Modeler IDE

Localization process in the Business Modeler IDE

1. **Set the languages** that your template supports.

2. **Set display names** for custom objects or override the display names for COTS (standard) objects.

3. **Add the Localization button to properties** so that localization administrators can enter localized text for the property values.

4. Install your template to a test server to verify the display names in the user interface, then install the template to a production server. When the template is installed to a production server, the display name text appears in the user interface for Teamcenter end users.

   > Caution:
   >
   > Do not use live update to place localization changes on a production server. Doing so could result in the following error:

```
Error Code: 515062  Error Message: Class referenced
```

Instead of using live update, always install localized templates using TEM or Deployment Center.

Setting language support

Introduction to setting language support

You can set the languages that a template supports at the time you create a project in the Business Modeler IDE, as well as by adding localization files to the project. Use the **SiteMasterLanguage** global constant to set the master locale for the template. Use the **Fnd0SelectedLocales** global constant to store the supported languages.

Note:

Translating text in a user interface is often called *localization* because it means translating text into languages used in different locales around the world.

Set the languages the template supports

In the Business Modeler IDE, you can set the languages that your template supports at the time you create the template project.

1. Choose **File→New→New Business Modeler IDE Template Project**.

2. Click **Next** until you get to the **Locales Selector** dialog box. Use this dialog box to select the languages your template supports. The displayed names of the data model you create in the template are viewable in these languages in the Teamcenter end-user interface.

   Caution:

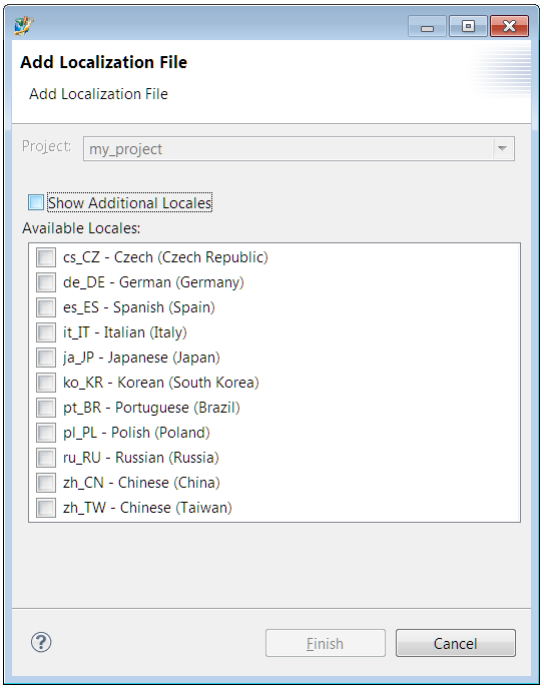
   Only select the locales your database supports.

Change the languages the template supports

In the Business Modeler IDE, the supported languages selected when the project is created are stored in the **Fnd0SelectedLocales** global constant. To see this constant, choose **BMIDE→Editors→Global Constants Editor**. To change the supported languages:

1. Open the **Project Files\extensions** folders, right-click the **lang** folder, and choose **Organize→Add localization files**.

   The **Add Localization File** dialog box is displayed.



2. Select the locales you want the template to support and click **Finish**.

   The **Fnd0SelectedLocales** global constant is updated.

Set the master locale for the template

In the Business Modeler IDE, use the **SiteMasterLanguage** global constant to set the master locale for the template. When you install a template to a server with this global constant set to a certain language, that language is the main language used for data input on the server.

When you click the **Localization** button in the Teamcenter client user interface, the resulting **Language Translations** dialog box displays the master locale setting in the **Master Locale** box.

1. On the menu bar, choose **BMIDE→Editors→Global Constants Editor**.

2. Select the **SiteMasterLanguage** constant in the **Global Constants** table.

3. Click the **Edit** button.

4. Enter a new language code. The default value is **en_US**. Valid values are the following *language_locale* codes:

cs_CZ	Czech as spoken in the Czech Republic		
de_DE	German as spoken in Germany	**ko_KR**	Korean as spoken in Korea
en_US	English as spoken in the United States	**pl_PL**	Polish as spoken in Poland

es_ES	Spanish as spoken in Spain	**pt_BR**	Portuguese as spoken in Brazil
fr_FR	French as spoken in France	**ru_RU**	Russian as spoken in Russia
it_IT	Italian as spoken in Italy	**zh_CN**	Chinese as spoken in China
ja_JP	Japanese as spoken in Japan	**zh_TW**	Chinese as spoken in Taiwan

5. Click **Finish**.

Users working in a configured alternate locale can modify master locale values from the alternate locale.

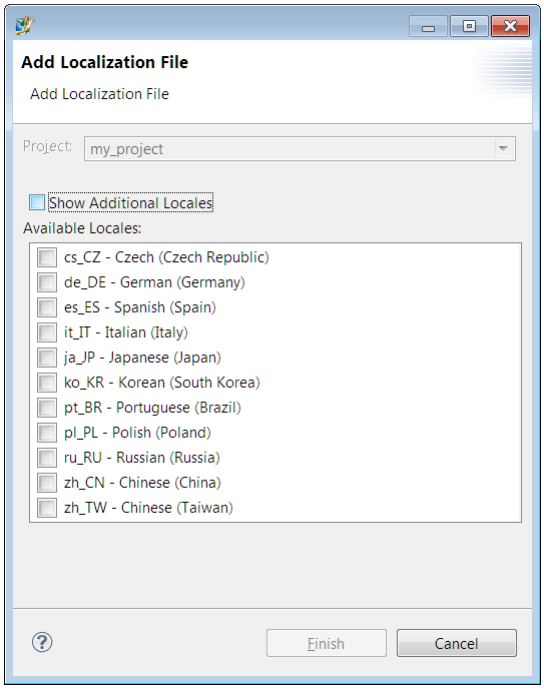
Add a locale

If the language you want to use is not on the list of available locales offered by Teamcenter, you can add a new locale using the Business Modeler IDE. For example, if you want to provide translations in Swedish, you can add the **sv_SE** locale to your template and to the server database.

1. Add the new locale to your template.

   a. In the Business Modeler IDE, open the **Project Files\extensions** folders, right-click the **lang** folder, and choose **Organize→Add localization files**.

      The **Add Localization File** dialog box is displayed.



   b. Select the **Show Additional Locales** check box.

c. Select the new locale you want to use. For example, if you want to use the Swedish locale, select **sv_SE - Swedish (Sweden)**.

d. Click **Finish**.

A message is displayed stating that Teamcenter does not support the locale by default and that you must add support for the new locale to the server.

A new *language_locale* folder is created in the **Project Files\extensions\lang** folder. To see the new file, right-click the view and choose **Refresh**.

The new locale is added to the list of available locales in the template, and is also added in the **Fnd0SelectedLocales** global constant. To see the new locale in the constant, choose **BMIDE→Editors→Global Constants Editor**.

e. Install the template to the server.

2. Add the new locale to server files.

a. Add the new locale to the **textsrv_text.xml** file.

A. Open the *TC_ROOT***lang\textserver\no_translation\textsrv_text.xml** file.

> Note:
>
> You can also use *TC_MSG_ROOT* in place of *TC_ROOT***lang**.

B. Determine the encoding that must be used for the new locale.

For the example of Swedish, the **textsrv_text.xml** file makes it easier to determine the encoding because it contains comments that show the encodings needed for Swedish and some other languages. To see the comments, search for the word **Swedish** in the file. The comments show that the following encodings must be used for Swedish:

ISO8859_1
ISO8859_4
ISO8859_15
NT_1254
UTF-8

> Note:
>
> Swedish is not listed in the comments for UTF-8 encoding because all locales are supported in UTF-8.

To determine the encodings to use for your new locale, consult the following tables.

Encodings for Oracle databases

Encoding	Oracle encoding name
ASCII	us7ascii
BIG5	zht16big5
BIG5	zht16mswin950
EUC	ja16euc
GB2312	zhs16cgb231280
GB2312	zhs16gbk
ISO8859_1	we8iso8859p1
ISO8859_2	ee8iso8859p2
ISO8859_4	nee8iso8859p4
ISO8859_5	cl8iso8859p5
ISO8859_6	ar8iso8859p6
ISO8859_7	el8iso8859p7
ISO8859_8	iw8iso8859p8
ISO8859_15	we8iso8859p15
KSC5601	ko16ksc5601
NT_852	ee8pc852
NT_862	iw8pc1507
NT_866	ru8pc866
NT_1250	ee8mswin1250

Encoding	Oracle encoding name
NT_1251	cl8mswin1251
NT_1252	we8mswin1252
NT_1253	el8mswin1253
NT_1254	tr8mswin1254
NT_1255	iw8mswin1255
NT_1256	ar8mswin1256
NT_1257	blt8mswin1257
SJIS	ja16sjis
UTF-8	utf8
UTF-8	al32utf8

Encodings for MS SQL databases

Encoding	MS SQL encoding name
BIG5	Chinese_Taiwan_Stroke_BIN
BIG5	Chinese_PRC_BIN
KSC5601	Korean_Wansung_BIN
NT_1250	Croatian_BIN
NT_1250	Czech_BIN
NT_1250	Hungarian_BIN
NT_1251	Cyrillic_General_BIN

Encoding	MS SQL encoding name
NT_1252	Latin1_General_BIN
NT_1253	Greek_BIN
NT_1254	Turkish_BIN
NT_1255	Hebrew_BIN
NT_1256	Arabic_BIN
NT_1257	Estonian_BIN
NT_1257	Latvian_BIN
NT_1257	Lithuanian_BIN
NT_1257	Polish_BIN
SJIS	Japanese_BIN

For the Swedish example, the encoding possibilities are:

Oracle database	MS SQL database	Windows Server required machine encoding	Linux Server required machine encoding
we8iso8859p1	latin1_general_bin	NT_1252	ISO8859-1
nee8iso8859p4	-	-	ISO8859-4
we8iso8859p15	-	-	ISO8859-15
tr8mswin1254	turkish_bin	NT_1254	-
utf8 / al32utf8	-	-	UTF-8

C. After you determine the proper encodings to use, add the new locale to the encodings **textsrv_text.xml** file.

For example, if you want to add Swedish, add the **sv_SE** locale to the following encoding lines in the file:

```
<key id="textsrv_dbEncodingName_ISO8859_1">ISO8859-1</key>
<key id="textsrv_encodingLocales_ISO8859_1">en_US;fr_FR;de_DE;it_IT;es_ES;sv_SE</key>
<key id="textsrv_dbEncodingName_ISO8859_4">ISO8859-4</key>
<key id="textsrv_encodingLocales_ISO8859_4">en_US;de_DE;sv_SE</key>
<key id="textsrv_dbEncodingName_ISO8859_15">ISO8859-15</key>
<key id="textsrv_encodingLocales_ISO8859_15">en_US;fr_FR;de_DE;it_IT;es_ES;sv_SE</key>
<key id="textsrv_dbEncodingName_NT_1254">NT-1254</key>
<key id="textsrv_encodingLocales_NT_1254">en_US;fr_FR;de_DE;it_IT;es_ES;sv_SE</key>
<key id="textsrv_dbEncodingName_UTF8">UTF-8</key>
<key
id="textsrv_encodingLocales_UTF8">cs_CZ;de_DE;en_US;es_ES;fr_FR;it_IT;ja_JP;ko_KR;ru_RU;zh_CN
;zh_TW;sv_SE</key>
<key id="locale_validation_encoding_sv_SE">iso-8859-1</key>
```

b. Add the new locale to the *TC_ROOT*\lang\textserver\en_US\textsrv_text_locale.xml file to ensure that the language display names appear when connecting to the rich client in English. Add the following line:

```
<key id="textsrv_localeName_language_locale">language-description</key>
```

For the Swedish example, add the following line to the file:

```
<key id="textsrv_localeName_sv_SE">Swedish</key>
```

c. If you want all the server-supplied user interface text and messages to appear in the new locale, copy the *TC_ROOT*\lang\textserver\en_US\ directory to a new locale directory and translate all the files into the new language.

For the Swedish example, copy the directory to a *TC_ROOT*\lang\textserver\sv_SE\ directory and translate the contents of the files to Swedish.

d. If you are using shared memory, you must remove the shared memory backing store files to ensure that all the modifications are considered when the pool manager and the middle tier are restarted.

The **TC_SHARED_MEMORY_DIR** environment variable value specifies the directory where the store backing files are stored.

e. If you updated the data in the text server file and you have enabled the rich client cache using TEM, you must update the client cache by running the **bmide_generate_client_cache** utility.

3. Verify the new locale in the rich client.

a. Add **-nl** *language_location* to the rich client startup script.

   For the Swedish example, add **-nl sv_SE** to the script, for example:

   ```
   TC_ROOT\portal\portal.bat -nl sv_SE
   ```

b. If you did not copy the *TC_ROOT***lang\textserver\en_US** directory to your new locale and translate all the contents of the files to the new language, the following error message is displayed:

   ```
   Unable to connect with the requested locale language_location.
   The application will exhibit mixed localizations because
   Teamcenter server is running in English.
   Details
   The Teamcenter server does not have 'language_location'
   localization files.
   ```

   If you did create the new directory with translated content, this error message does not appear.

c. To verify that the new locale is available for data input, click the **Localization** button to see the new locale include in the list of available locales.

   For example, in the Query Builder application, select a query in the **Saved Queries** pane and click the **Localization** button. Click in the **Locale** cell to see the new locale listed.

d. To change the language for inputting text to the new locale, enter the new locale to the **TC_language_data_entry** preference. To see these language preferences, choose **Edit→Options**, click the **Search** at the bottom of the **Options** dialog box, and search for preferences beginning with **TC_language**.

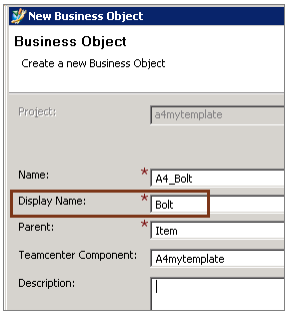
Users working in a configured alternate locale can modify master locale values from the alternate locale.

Setting display names

What is a display name?

A display name is the name that displays for data model objects in Teamcenter user interface clients. You can use a Business Modeler IDE to set the display name for your custom objects as well as override display names on COTS (standard) Teamcenter objects.

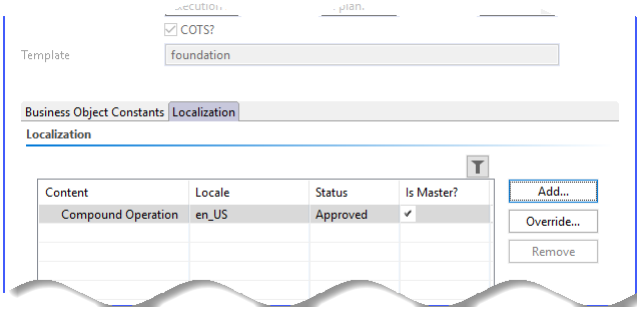
For example, if you create a new part business object type named **A5_Bolt**, you can use the **Display Name** box in the creation dialog box to set the display name as **Bolt**.



Add a display name for the object in other languages

1. Open the new business object type.

2. To the right of the **Localization** table, click **Add**.



3. In the **Localization** dialog box, select the locale and enter the display name for the locale.

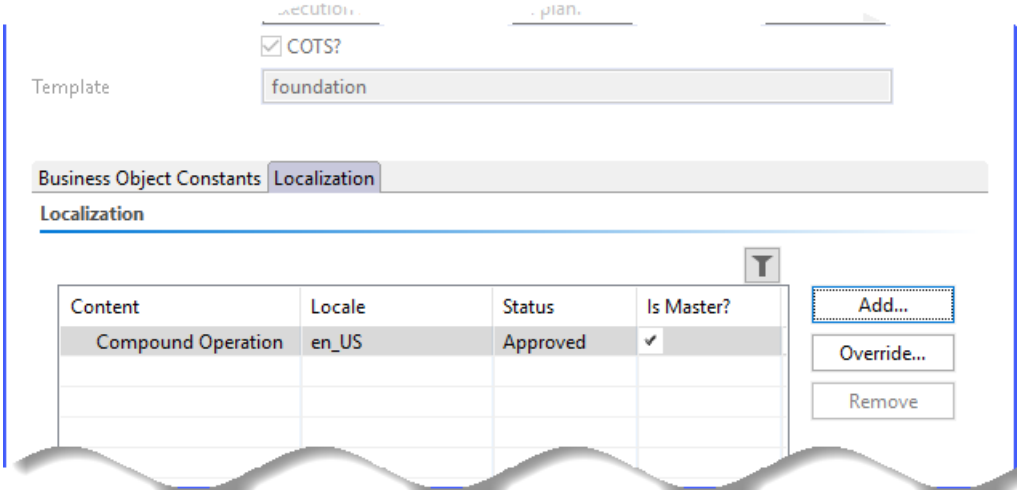
> Caution:
>
> If you change the display name of an object in the master locale, you must also change the display names for the object in all locales. If you do not, the status of display names in other locales is marked as invalid.

Change display names for business objects and option types

You can change the display name of any business object, as well as the following object types: ID contexts, note types, occurrence types, statuses, and view types.

1. In the Business Modeler IDE, open a custom object or a COTS (standard) object.

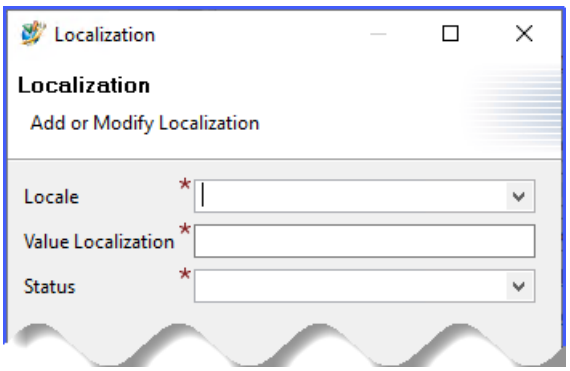
2. Open the **Localization** tab.



3. In the **Localization** table, select a row and click a command at the right side of the table:

Use this command	To do this
Add	Add a new display name.
Override (For a COTS object)	Change the selected display name.
Edit (For a custom object or an already overridden COTS object)	Change the selected display name.
Remove	Remove the selected display name.

The **Localization** dialog box appears.



4. In the **Localization** dialog box, specify the options for the localization:

For this option	Do this
Locale	Select the language locale where the text is to be used.

For this option	Do this
	Available locales depend on the languages set in the global constant **Fnd0SelectedLocales**. **cs_CZ** Czech as spoken in the Czech Republic **de_DE** German as spoken in Germany **en_US** English as spoken in the United States **es_ES** Spanish as spoken in Spain **fr_FR** French as spoken in France **it_IT** Italian as spoken in Italy **ja_JP** Japanese as spoken in Japan **ko_KR** Korean as spoken in Korea **pl_PL** Polish as spoken in Poland **pt_BR** Portuguese as spoken in Brazil **ru_RU** Russian as spoken in Russia **zh_CN** Chinese as spoken in China **zh_TW** Chinese as spoken in Taiwan
Value Localization	Type the value for the display name.
Status	Select the status of the text change in the approval life cycle: **Approved** The text change is approved for use. **Invalid** The text change is not valid. **Pending** The text change is pending approval. **Review** The text change is in review. Tip: You can export text for translation based on its status using the **l10n_import_export** utility.

5. Click **Finish**.

Set display names for properties

Properties display information about objects in Teamcenter, such as name, creation date, owner, and so on. Using the Business Modeler IDE, you can define the name that displays for properties in the Teamcenter user interface.

1. In the Business Modeler IDE, open a business object and click the **Properties** tab.

2. Select a string property in the table for which you want to change the display name, for example, **object_name**.

3. Click the **Override**, **Edit**, or **Add** buttons to the right of the **Localization** table to change the text.

> Note:
>
> To be proficient with properties, you need to know both the internal name of the property and its display name. You can change the settings in the rich client to display the internal name of a property in the user interface. Log on to the rich client as an administrator, choose **Edit**>**Options**, and in the left pane of the **Options** dialog box, choose **Options**>**General**>**UI**. In the right pane, click the **Sys Admin** tab and select **Real Property Name**. To verify the change, select an item in the rich client and choose **View**>**Properties**.

> Tip:
>
> You can change how values for properties are displayed in the Active Workspace user interface by using property formatters.

Set display names for lists of values (LOVs)

Lists of values (LOVs) are the pick lists displayed in Teamcenter when end users click an arrow in a data entry box. Using the Business Modeler IDE, you can define the text that displays for string LOVs in the Teamcenter user interface.

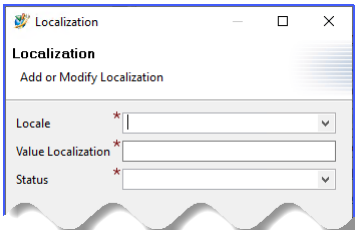
1. In the Business Modeler IDE, open the **Extensions\LOV** folders.

2. Open an LOV and select a value in the LOV table.

3. Click the **Localization** button to the right of the table.

   The **LOV Value Localization** dialog box is displayed.

   The **Localization** dialog box appears.



4. In the **LOV Value Localization** dialog box, click the **Override**, **Edit**, or **Add** button.

   The **Localization** dialog box is displayed.

5. Perform the following in the **Localization** dialog box:

   a. In the **Value Localization** box, type the value as you want it to display in the user interface.

b. Click the arrow in the **Locale** box to select the language locale where the text is used.

c. If a description was previously entered for the LOV value, in the **Description Localization** box, type a description for the display text.

d. In the **Status** box, select the status of the text change in the approval life cycle.

e. Click **Finish**.

6. Click **Finish** in the **LOV Value Localization** dialog box.

Validating localizations

The display names (localizations) entered in the Business Modeler IDE are validated to ensure that the characters used in the display names are valid for a given locale. The same validations are done by the Business Modeler IDE parser whenever you open, reload, or import a Business Modeler IDE project to the Business Modeler IDE client.

To achieve this validation, the Business Modeler IDE now installs the **textsrv_text.xml** file into the *TC_ROOT***/lang/textserver/no_translation** directory.

The **textsrv_text.xml** file contains encodings per locale that are used by the Business Modeler IDE client to validate the display names. This file contains many XML entries, but the following sample code shows how encodings are stored per locale:

```
<!-- SECTION DEFINING THE SMALLEST OR CUSTOM ENCODING FOR EACH
LOCALE: THIS IS USED FOR BMIDE LOCALIZATION VALIDATION -->
<key id="locale_validation_encoding_en_US">us-ascii</key>
<key id="locale_validation_encoding_cs_CZ">iso-8859-2</key>
<key id="locale_validation_encoding_pl_PL">iso-8859-2</key>
<key id="locale_validation_encoding_de_DE">iso-8859-1</key>
<key id="locale_validation_encoding_es_ES">iso-8859-1</key>
<key id="locale_validation_encoding_fr_FR">iso-8859-1</key>
<key id="locale_validation_encoding_it_IT">iso-8859-1</key>
<key id="locale_validation_encoding_pt_BR">iso-8859-1</key>
<key id="locale_validation_encoding_ja_JP">shift_jis</key>
<key id="locale_validation_encoding_ko_KR">euc_kr</key>
<key id="locale_validation_encoding_ru_RU">iso-8859-5</key>
<key id="locale_validation_encoding_zh_CN">gb2312</key>
<key id="locale_validation_encoding_zh_TW">big5</key>
```

In the XML sample code, not only are the encodings stored per locale, they are available only for the locales included with standard Teamcenter. Currently, each locale is set to its minimum encoding. For example, the encoding for **en_US** is **US-ASCII**. This means if you enter display names (localizations) in the Business Modeler IDE for **en_US**, the Business Modeler IDE validates that the characters entered in the **Display Name** box are valid in the **US-ASCII** character set. The same logic applies to localizations entered in other locales.

The encodings supplied in the **textsrv_text.xml** file are used *only* by the Business Modeler IDE client for validations on localizations during the loading of a Business Modeler IDE project or when users enter display names in the user interface. These encodings are not used during template deployment. When you install your template to the database, the settings in the **textsrv_text.xml** file are not used. Instead, the Business Modeler IDE deploy utilities use the encoding set for the database and the server host.

Encodings in the **textsrv_text.xml** file are the minimum encodings. You may use a larger encoding because your database supports it. For example, you may want to use the Euro symbol in **en_US** files because your database encoding is set to **ISO-8859-15**. In such cases, you can modify the entries in the **textsrv_txt.xml** file and provide your larger encoding for each of the locales. For example, you change the encoding for **en_US** as **ISO-8859-15**:

```
<key id="locale_validation_encoding_en_US">iso-8859-15</key>
```

You can change the encoding for each of the locales (shown in the previous code example) to use an encoding that is supported in your database. The Business Modeler IDE then uses your custom encoding to validate the localizations in the Business Modeler IDE client.

Teamcenter includes localizations only for a certain list of locales. For example, Teamcenter does not include localization for the Romanian (**ro_RO**) locale. You may be supporting the Romanian locale and if you want the Business Modeler IDE client to validate all localizations entered in the Romanian locale, you must modify the **textsrv_text.xml** file and add an entry for this locale. For example, you can add the following:

```
<key id="locale_validation_encoding_ro_RO">iso-8859-16</key>
```

Add the Localization button to properties

The **Localization** button allows localization administrators to enter localized text for the property values. To place the **Localization** button on properties in the rich client, use the Business Modeler IDE to set the **Localizable** property constant to **true** for those properties.

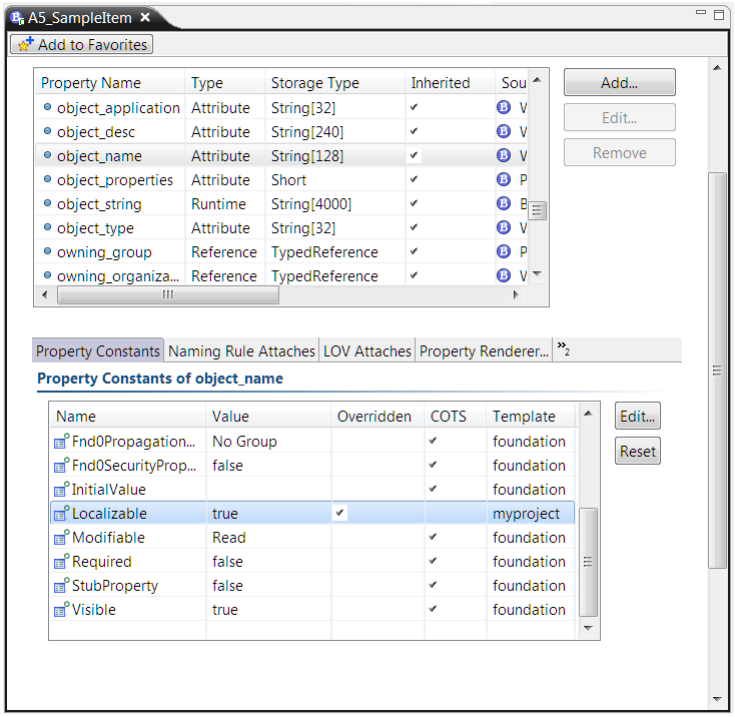
> Note:
>
> A system administrator must use the Organization application to add localization administrators to a translator group to give them authorization to enter the localized text.

1. In the Business Modeler IDE, open a business object and click the **Properties** tab.
2. Select a string property in the table, for example, **object_name**.
3. In the **Property Constants** table, select the **Localizable** property constant.
4. Click the **Edit** button to the right of the **Property Constants** table.

   The **Modify Property Constant** dialog box appears.

5. Click the arrow in the **Value** box to change the value to **true**.

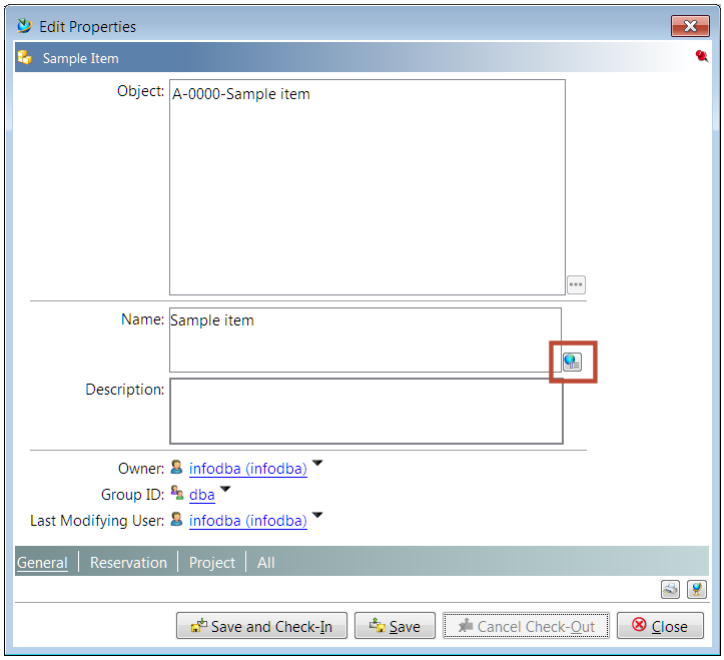
6. Click **Finish**.



7. Package your template and install it using Teamcenter Environment Manager (TEM) or Deployment Center.

8. Verify the **Localization** button appears on the property in the rich client.

   For example, if you want to add the **Localization** button to the **object_name** property on the **Item** business object, open an **Item** business object and click the **More Properties** link in the **Summary** tab. The **Localization** button appears to the right of the **Name** box.



Note:

To remove the **Localization** button from this property, in the Business Modeler IDE, select the **Localizable** property constant for the property and click the **Reset** button. Then deploy the change.

If you add the **Localization** button to a property by setting the **Localizable** property constant to **true**, and then later decide to remove the button by setting the constant to **false**, you must run the **l10n_purge_translations** utility to remove translations that were entered using the **Localization** button on that property. The utility must be run after the Business Modeler IDE template is deployed. The utility may be run on a live database, but for maximum efficiency, it is best to run the utility while other users are not logged on.

The **l10n_purge_translations** utility is necessary only if the **Localizable** constant on a property is moved or deleted from one level, but still exists at another level of the hierarchy. For example, if the **Localizable** property constant is set to **false** on a property on a business object, and there are no sub-business objects that need to be set to **true**, the utility does not need to be executed.

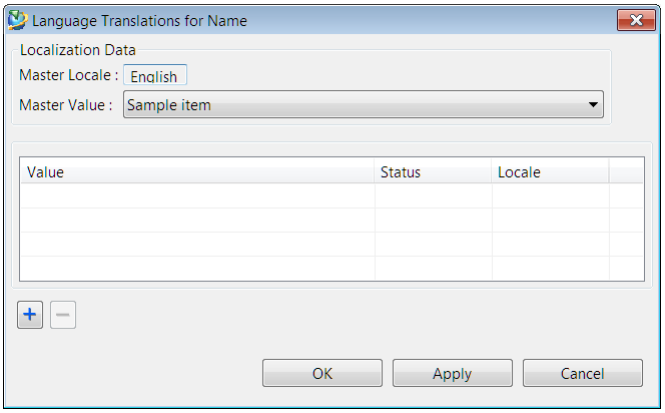
Caution:

The **Localization** button appears when the property is displayed in the **Properties** dialog box, **Summary** view, and **Viewer** view.

However, when a suggestive LOV is attached to a property and the end user enters a custom entry in the rich client (which is allowed for suggestive LOVs), the **Localization** button is not displayed. If translations are required for this custom entry, the custom entry must be added as a value to the LOV in the Business Modeler IDE and translations must be added to the LOV.

9. To enter localized text, click the **Localization** button.

The **Language Translations for Name** dialog box is displayed.



10. Click the **+** button to add localization text.

Migrate a custom template to the newer language framework

If you have an older custom template created prior to Teamcenter 8.2, you must migrate the template to the newer framework for language support using the following process:

1. After upgrading the older custom template to the newer data model, **migrate the text for properties and relations** to the newer language framework.

2. **Create a default set of localization files** for the upgraded template project.

Migrate property and relation names

You can migrate the text for property and relation names from a project previous to Teamcenter 8.2 to the new language framework. The Property Name and Relation Name Migration wizard reads the text from the text server XML files and creates the display name for the custom property names and relation names.

Previous to Teamcenter 8.2, property and relation names text in different languages was stored in the *TC_ROOT***lang\textserver***language_locale* directory in the **user_property_names.xml** and **system_property_names_locale.xml** files. Beginning in Teamcenter 8.2, property and relation names text can be stored in the database.

> Note:
>
> If a text server file defines an empty string as the display name, the migration routine sets the **Visible** property constant to **false** to make the property invisible.

1. Upgrade the older custom template to the newer data model using one of the following methods:

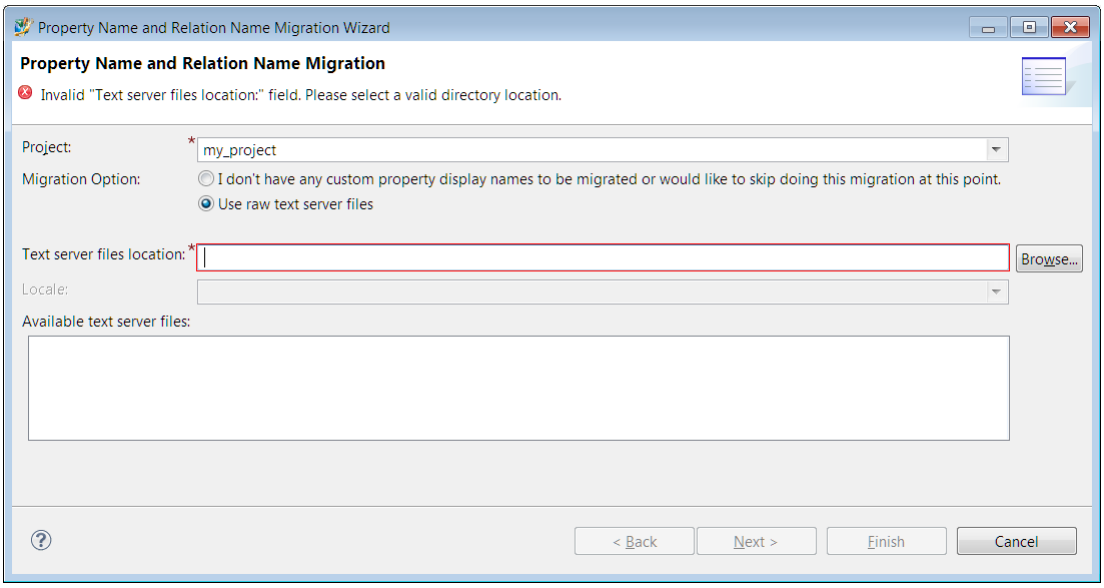
- If your template project is not already in the workspace, import it into the new version of the Business Modeler IDE by choosing **File→Import→Business Modeler IDE→Import a Business Modeler IDE Template Project**.

- If your template project is already in the workspace, upgrade it to the new version of the Business Modeler IDE by choosing **BMIDE→Upgrade Tools→Re-run Template Project Upgrade**.

> Note:
>
> If no text server text was assigned to some custom objects in the upgraded template, now that the template is upgraded, you can add display names to these custom objects manually or use the **Default Localization Creation** wizard.

2. Select the upgraded project and choose **BMIDE→Upgrade Tools→Property Name and Relation Name Migration Wizard**.

   The Property Name and Relation Name Migration wizard runs.



3. In the **Property Name and Relation Name Migration** dialog box, perform the following steps:

   a. Ensure that the **Project** dialog box displays the selected upgraded project.

   b. In the **Migration Option** area, click the **Use raw text server files** button.

      (Click the other button only if you decide not to continue with the migration process. The other button is titled **I don't have any custom property names to be migrated or would like to skip doing this migration at this point.**)

c. Click the **Browse** button to the right of the **Text server files location** box and browse to the directory for the text server files on the source (older) Teamcenter server, for example, *TC_ROOT***lang\textserver**. This value is also provided by the value of the **TC_MSG_ROOT** environment variable.

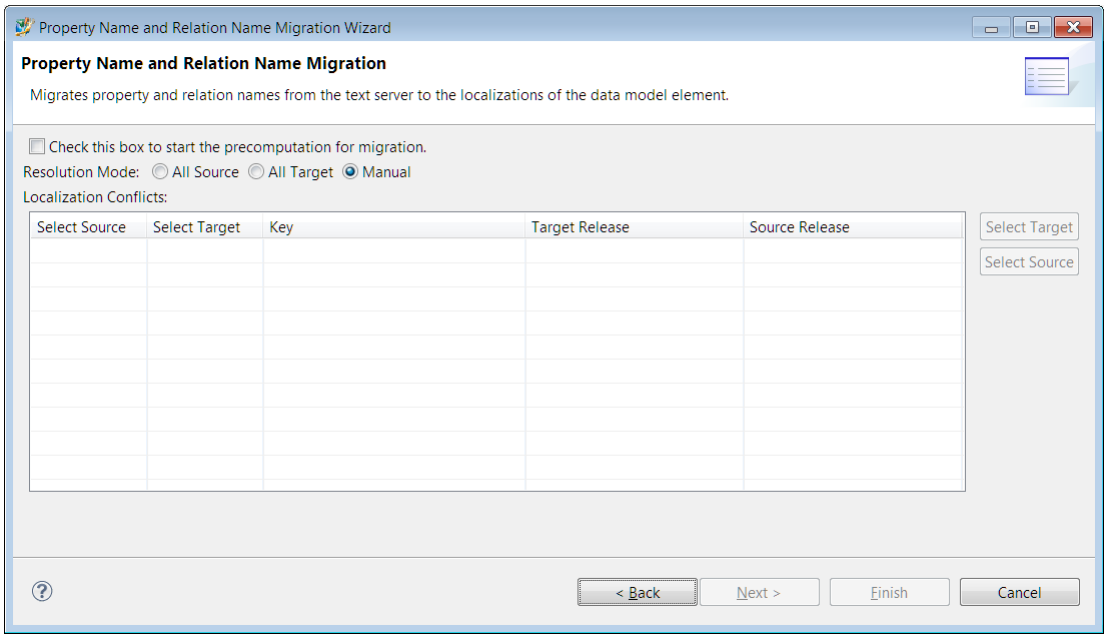
> Note:
>
> The migration wizard takes only one location from which to migrate the localizations. However, some users may have user-level, text server localizations in the **TC_USER_MSG_DIR**directory. Therefore, if users have user-level, text-server localizations in the *TC_USER_MSG_DIR**language_locale***user_property_names.xml** file, before running the migration wizard, you must merge the contents of this file with the one located in the **TC_MSG_ROOT** location. (Replace *language_locale* with the value of the **TC_language_default** environment variable, for example, **en_US**.)

d. Click the arrow in the **Locale** box to choose the language you want to migrate.

> Note:
>
> After the migration for this locale is complete, repeat the migration for the other locales if there are conflicts in other locales.

e. Click **Next**.

The next dialog box in the migration wizard is displayed.



4. Perform the following in the next **Property Name and Relation Name Migration** dialog box:

a. Select the **Check this box to start the precomputation for migration** check box. This parses the text server XML files and processes the keys.

   If there are any conflicts after precomputation, they are displayed in the **Localization Conflicts** table.

   Conflicts may appear when comparing source (from the older version of Teamcenter) with target (from the newer version of Teamcenter) because some text is updated between Teamcenter versions.

b. If the **Localization Conflicts** table displays conflicts, select whether you want to use the source or target localization text. You can click the **All Source** button to use all the text from the older version of Teamcenter, or click the **All Target** button to use all the text from the most recent version of Teamcenter, or click the **Manual** button to identify them one at a time using the **Select Target** or **Select Source** buttons to the right of the table.

   If there are no customized entries listed in the **Localization Conflicts** table, click the **All Target** button.

c. When there are no conflicts, click the **Finish** button.

5. Open the **Console** view to see the path to the log file that contains the summary of the migration.

6. After the property names and relation names are migrated, run the Default Localization Creation wizard to create the default display names.

Create a default localization

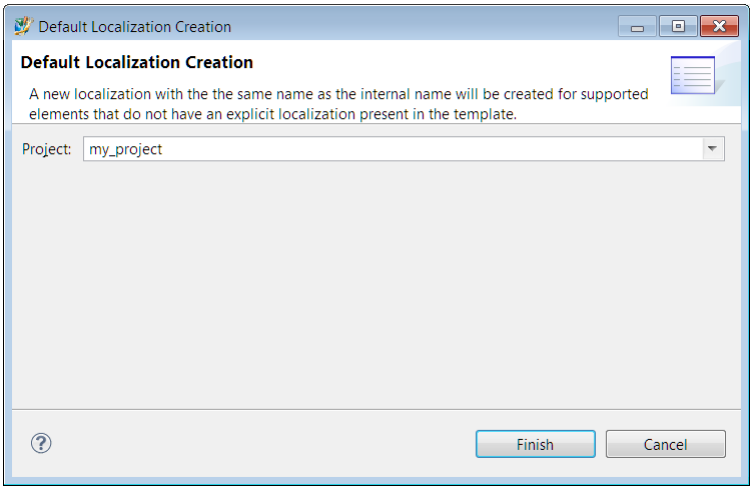
Run the Default Localization Creation wizard in the Business Modeler IDE to create a default set of localization files for an upgraded project.

1. After upgrading a pre-Teamcenter 8.2 project to the newer localization framework, migrate the text for property and relation names using the Property Name and Relation Name Migration wizard.

2. Select the upgraded project and choose **BMIDE→Upgrade Tools→Default Localization Creation Wizard**.

   Click **Next**.

   The Default Localization Creation wizard runs.



3. Click **Finish**.

   This creates the necessary directory structure and the empty language files to store the display names. The localization files are located in the **Project Files\extensions\lang** folder.

4. Open the **Console** view to see the path to the log file that contains the summary of the action.

Import localization files into your custom template

If you have entered translated text into user interface localization files, then you can import the localization files to your template using the Business Modeler IDE.

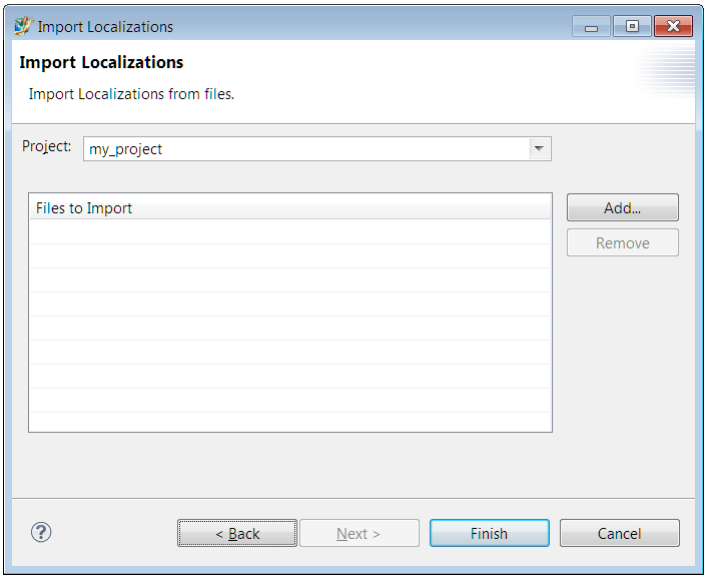
> Note:
>
> To extract user interface text in bulk for translation into different languages, export the text into localization files using one of these methods:
>
> - Run the **l10n_import_export** utility.
> - In the rich client, choose **Tools→Localization→Export Translations**.

1. In the Business Modeler IDE, choose **File→Import**.

2. In the **Select** dialog box, choose **Business Modeler IDE→Import Localizations**.

3. Click **Next**.

   The Import Localizations wizard is displayed.



4. Perform the following steps in the **Import Localizations** dialog box:

   a. In the **Project** box, click the arrow and select the project to receive the localization files.

   b. Next to the **Files to Import** table, click **Add** to browse to the location of the localization files and select them.

   c. Click **Finish**.

Localization and live update

> Caution:
>
> Do not use live update to place localization changes on a production server. Instead of using live update, install localized templates using Teamcenter Environment Manager (TEM) or Deployment Center.

Because localization changes can contain changes to the schema, and schema changes cannot be live updated to a production server, attempting to use live update to deploy localization changes could result in the following error:

```
Error Code: 515062  Error Message: Class referenced
```

This error occurs in a couple of notable situations.

Example:

You want to add the **Localization** button to a property, so you set the **Localizable** property constant to **true**. Usually setting a constant value does not result in changes to any classes or attributes. However, in this case, an additional hidden attribute is created on the class used by the associated business object.

- On the **Item** business object, on the **object_desc** or **object_name** property, you set the **Localizable** property constant to **true** and then attempt to live update the change. The 515062 error occurs.

Example:

To make a list of values available for setting a property value in the client, in the Business Modeler IDE you attach an LOV to a property. The attachment results in a hidden attribute being placed on the source business object of the property. If you attempt the live update the change, the 515062 error occurs.

A. Non-English locale environment settings

Verify that your locale is supported

On Windows systems, if you do not use UTF-8, ensure the locale you want to use is supported on your host. Perform the following steps to set the Windows system locale and install the required language packs:

1. Open the **Regional and Language Options** dialog box in the Windows Control Panel.
2. In the **Languages** tab, set the required language for the menus and dialog boxes.
3. In the **Region and Language** dialog box, click the **Administrative** tab.
4. Under **Language for non-Unicode programs**, click **Change system locale**.
5. In the **Region and Language Settings** dialog box, verify the correct locale (language and country) is selected. If not, choose the correct locale.
6. Close all dialog boxes and restart your system to install and configure the required language pack.

On Linux systems, to verify that the desired locales are supported on your host, type the following command:

```
locale -a
```

This command returns a list of all locales the host supports. If a locale you need is not included in the list, contact your system administrator to install the required language pack.

Keep in mind that some Linux platform GUIs may allow you to set a locale that is not in the list of supported locales on the host. Make sure the locale you set is supported on the host.

To verify that a desired character set is available on your host, type the following command, which lists character sets supported on the host:

```
locale charmap
```

Configuring a UTF-8 environment for Teamcenter

Overview of UTF-8 configuration

Teamcenter supports the Unicode UTF-8 character set on Windows and Linux hosts that are configured to process UTF-8.

To configure your Teamcenter host to use Unicode UTF-8, perform the following steps before you install Teamcenter:

1. Configure your operating system to run Unicode UTF-8.

2. Install a database server and enable Unicode UTF-8 character set support during installation.

To configure your Teamcenter host to use the UTF-8 character set, install a database server and enable UTF-8 character set support during installation.

Set the required values for your platform, locale, and database type *before* you begin installing Teamcenter.

Enable UTF-8 support for Teamcenter servers and clients during Teamcenter installation

- **Teamcenter servers**

  With UTF-8 support configured on your host, can create a UTF-8-enabled Teamcenter database during Teamcenter installation.

- **Two-tier rich client**

  If the Teamcenter database is configured for the UTF-8 character set, **UTF8** is selected by default in the **TcServer Character Encoding Settings** panel in TEM.

- **Four-tier rich client**

  When installing the Teamcenter web tier, in the **TcServer Character Encoding Settings** panel in TEM, select **UTF8**.

  The four-tier rich client can run on any Windows or Linux platform running any language character set.

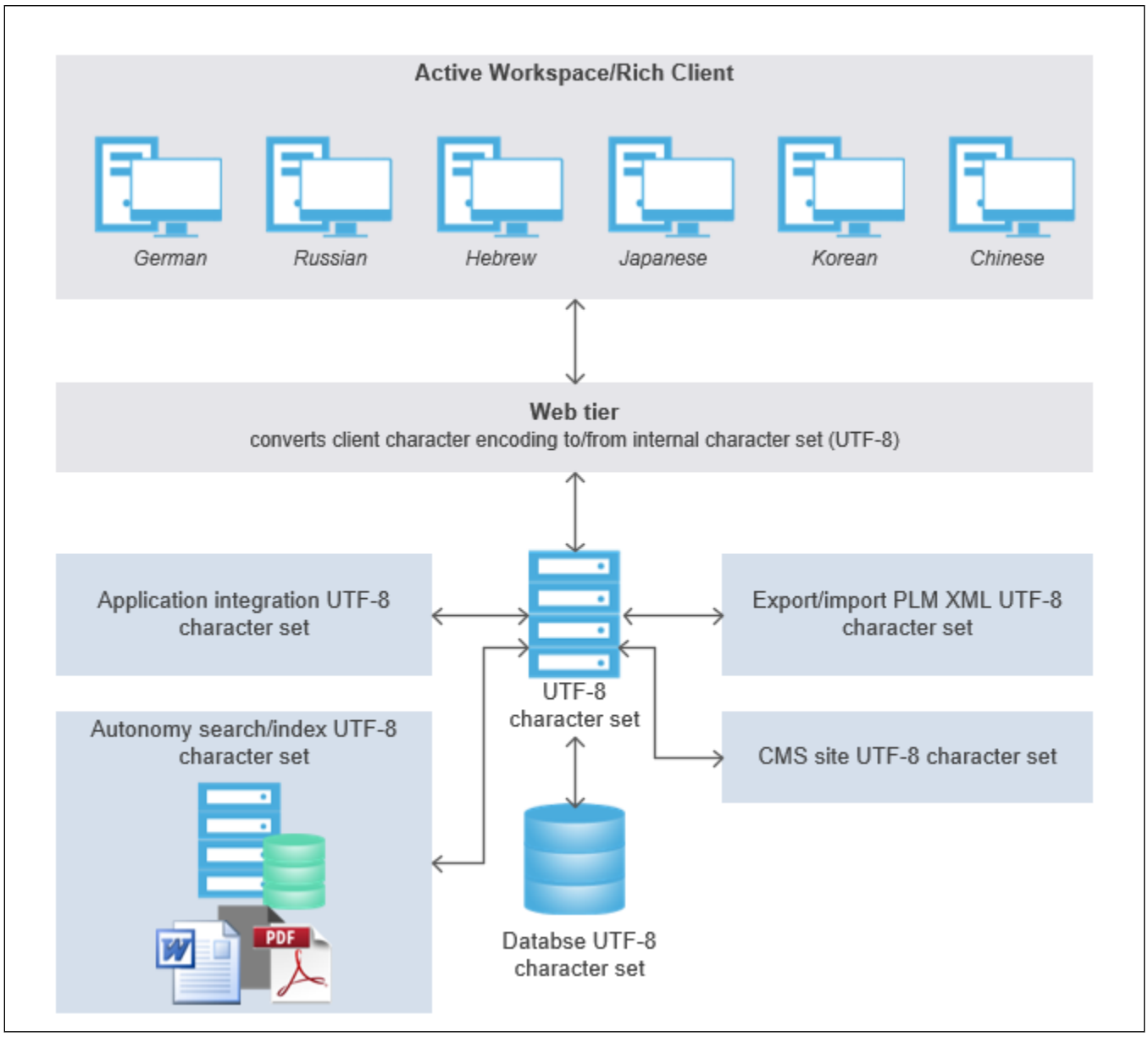
- **Web tier**

  Make sure UTF-8 support is configured on the web tier host.

  The web tier can run on any Windows or Linux platform running any language character set. The Teamcenter web tier converts client character encoding to and from UTF-8 as it passes through the web tier.

The following example shows a Teamcenter configuration for restricted Unicode UTF-8 character set support with clients displaying multiple locales. Servers in this configuration run a Unicode UTF-8 character set operating system.

On Windows platforms, if the database is configured for the UTF-8 character set, the Teamcenter server operates in UTF-8 mode independent of the system locale.



Unicode homogeneous server platform configuration

- Your Linux platform administrator must configure the host to run the Unicode UTF-8 character set operating system by default. This enables all software running on in the operating system environment to recognize the default character set is UTF-8.

- Teamcenter does not support Unicode Supplementary Characters.[1]

1 Unicode Supplementary Characters are characters in the Unicode Character Standard outside of the Basic Multilingual Plane (BMP), that is, characters with code point values larger than 0xFFFF.

- If you import translated content in languages that require multibyte characters, such as Russian and Chinese Simplified, you must to ensure that titles and other properties display correctly in your environment.

Configure UTF-8 environment settings

If you use UTF-8, select the **al32utf8** or **utf8** character set when you install your database server.[2]

In addition, on Linux systems, set the **LANG** and **LC_ALL** system environment variables to the appropriate values for your locale and platform. These variables must have identical values to function properly.

Values for LANG and LC_ALL

Locale	Value
Chinese (Simplified)	**zh_CN.utf8**
Chinese (Traditional)	**zh_TW.utf8**
Czech	**cs_CZ.utf8**
English	**en_US.utf8**
French	**fr_FR.utf8**
German	**de_DE.utf8**
Hebrew	**he_IL.utf8**
Italian	**it_IT.utf8**
Japanese	**ja_JP.utf8**
Korean	**ko_KR.utf8**
Polish	**pl_PL.utf8**
Portuguese (Brazilian)	**pt_BR.utf8**
Russian	**ru_RU.utf8**
Spanish	**es_ES.utf8**

In Hebrew locales, set the following additional variables:

1. In the *TC_DATA*/**tc_profilevars** file, set **TC_XML_ENCODING** to **UTF-8**.

2. In two-tier environments, set **TC_CHARACTER_ENCODING_SET** to **UTF8** in the following files:

   - *TC_ROOT*/**tccs/Start_TcServer1**
   - *TC_ROOT*/**pool_manager/confs/MYDB/mgrstart**

2 Oracle recommends **al32utf8**. **UTF8** supports only supports Unicode Version 3.0 and earlier.

Do not set the **TC_XML_ENCODING** or **TC_CHARACTER_ENCODING_SET** environment variables in the system environment. TEM sets these values in the Teamcenter configuration.

Configuring a non-UTF-8 environment for Teamcenter

To ensure correct display and processing of Teamcenter data, set the required values in your operating system environment. Use the appropriate values for your locale and platform.

Environment settings on non-UTF-8 systems

		Value	
Locale	**Setting**	**Linux**	**Microsoft Windows**
Chinese (Simplified), GB2312-80 encoding	Database character set (Oracle)	**zhs16cgb231280** or **zhs16gbk**	**zhs16cgb231280** or **zhs16gbk**
	Database collation (MS SQL Server)[1]	N/A	**chinese_prc_bin**
	LANG and **LC_ALL**[2]	**zh_CN**	N/A
Chinese (Simplified), GBK encoding	Database character set (Oracle)	**zhs16cgb231280** or **zhs16gbk**	**zhs16cgb231280** or **zhs16gbk**
	Database collation (MS SQL Server)[1]	N/A	**chinese_prc_bin**
	LANG and **LC_ALL**[2]	**zh_CN.gb18030**	N/A
Chinese (Traditional)	Database character set (Oracle)	**zht16big5** or **zht16mswin950**	**zht16big5** or **zht16mswin950**
	Database collation (MS SQL Server)[1]	N/A	**chinese_taiwan_stroke_bin**
	LANG and **LC_ALL**[2]	**zh_TW**	N/A
Czech	Database character set (Oracle)	**ee8mswin1250**	**ee8mswin1250**
	Database collation (MS SQL Server)[1]	N/A	**czech_bin**
	LANG and **LC_ALL**[2]	**cs_CZ**	N/A
English	Database character set (Oracle)	**we8iso8859p1** or **we8iso8859p15**[3] or **we8mswin1252**[4]	**we8iso8859p1** or **we8iso8859p15**[3] or **we8mswin1252**[4]
	Database collation	N/A	**latin1_general_bin**

Notes:

1. The database collation you select during Microsoft SQL Server installation determines the database character set.
2. Set **LANG** and **LC_ALL** in the system environment variables. These variables must have identical values to function properly.
3. **we8iso8859p15** contains additional characters, including the euro symbol (€).
4. **we8mswin1252** contains more characters than **ISO-8859-15**.
5. No **ISO-8859-15** equivalent is available for this locale.
6. Siemens Digital Industries Software does not provide a Hebrew translation. The configuration settings shown allow data entry in Hebrew, but user interface text is in English.
7. If you migrate to **ko16ksc5601** from UTF-8, some data may be truncated. You must modify truncated valued because Teamcenter does not support modifying the default field size.

		Value	
Locale	**Setting**	**Linux**	**Microsoft Windows**
	(MS SQL Server)[1]		
	LANG and **LC_ALL**[2]	**en_US** or **en_US.iso885915**	N/A
French	Database character set (Oracle)	**we8iso8859p1** or **we8iso8859p15**[3] or **we8mswin1252**[4]	**we8iso8859p1** or **we8iso8859p15**[3] or **we8mswin1252**[4]
	Database collation (MS SQL Server)[1]	N/A	**latin1_general_bin**
	LANG and **LC_ALL**[2]	**fr_FR**[5]	N/A
German	Database character set (Oracle)	**we8iso8859p1** or **we8iso8859p15**[3] or **we8mswin1252**[4]	**we8iso8859p1** or **we8iso8859p15**[3] or **we8mswin1252**[4]
	Database collation (MS SQL Server)[1]	N/A	**latin1_general_bin**
	LANG and **LC_ALL**[2]	**de_DE**[5]	N/A
Hebrew[6]	Database character set (Oracle)	**iw8iso8859p8** or **iw8mswin1255**	**iw8iso8859p8** or **iw8mswin1255**
	Database collation (MS SQL Server)[1]	N/A	**hebrew_bin**
	LANG and **LC_ALL**[2]	**iw_IL.utf8**	N/A
Italian	Database character set (Oracle)	**we8iso8859p1** or **we8iso8859p15**[3] or **we8mswin1252**[4]	**we8iso8859p1** or **we8iso8859p15**[3] or **we8mswin1252**[4]
	Database collation (MS SQL Server)[1]	N/A	**latin1_general_bin**
	LANG and **LC_ALL**[2]	**it_IT**[5]	N/A
Japanese (EUC)	Database character set (Oracle)	**ja16euc**	**ja16euc**
	Database collation (MS SQL Server)[1]	N/A	N/A
	LANG and **LC_ALL**[2]	**ja_JP.eucjp**	N/A
Japanese (Shift-JIS)	Database character set (Oracle)	**ja16sjis**	**ja16sjis**
	Database collation (MS SQL Server)[1]	N/A	**japanese_bin**
	LANG and **LC_ALL**[2]	**ja_JP.sjis**	N/A

Notes:

1. The database collation you select during Microsoft SQL Server installation determines the database character set.
2. Set **LANG** and **LC_ALL** in the system environment variables. These variables must have identical values to function properly.
3. **we8iso8859p15** contains additional characters, including the euro symbol (€).
4. **we8mswin1252** contains more characters than **ISO-8859-15**.
5. No **ISO-8859-15** equivalent is available for this locale.
6. Siemens Digital Industries Software does not provide a Hebrew translation. The configuration settings shown allow data entry in Hebrew, but user interface text is in English.
7. If you migrate to **ko16ksc5601** from UTF-8, some data may be truncated. You must modify truncated valued because Teamcenter does not support modifying the default field size.

		Value	
Locale	**Setting**	**Linux**	**Microsoft Windows**
Korean	Database character set (Oracle)	**ko16ksc5601**[7]	**ko16ksc5601**[7]
	Database collation (MS SQL Server)[1]	N/A	**korean_wansung_bin**
	LANG and **LC_ALL**[2]	**ko_KR.EUC**	N/A
Polish	Database character set (Oracle)	**ee8mswin1250**	**ee8mswin1250**
	Database collation (MS SQL Server)[1]	N/A	**polish_bin**
	LANG and **LC_ALL**[2]	**pl_PL.ISO8859-2**	N/A
Portuguese (Brazilian)	Database character set (Oracle)	**we8iso8859p1** or **we8iso8859p15**[3] or **we8mswin1252**[4]	**we8iso8859p1** or **we8iso8859p15**[3] or **we8mswin1252**[4]
	Database collation (MS SQL Server)[1]	N/A	**latin1_general_bin**
	LANG and **LC_ALL**[2]	**pt_BR**[5]	N/A
Russian	Database character set (Oracle)	**cl8mswin1251** or **cl8iso8859p5**	**cl8mswin1251** or **cl8iso8859p5**
	Database collation (MS SQL Server)[1]	N/A	**cyrillic_general_bin**
	LANG and **LC_ALL**[2]	**ru_RU**	N/A
Spanish	Database character set (Oracle)	**we8iso8859p1** or **we8iso8859p15**[3] or **we8mswin1252**[4]	**we8iso8859p1** or **we8iso8859p15**[3] or **we8mswin1252**[4]
	Database collation (MS SQL Server)[1]	N/A	**latin1_general_bin**
	LANG and **LC_ALL**[2]	**es_ES**[5]	N/A

Notes:

1. The database collation you select during Microsoft SQL Server installation determines the database character set.
2. Set **LANG** and **LC_ALL** in the system environment variables. These variables must have identical values to function properly.
3. **we8iso8859p15** contains additional characters, including the euro symbol (€).
4. **we8mswin1252** contains more characters than **ISO-8859-15**.
5. No **ISO-8859-15** equivalent is available for this locale.
6. Siemens Digital Industries Software does not provide a Hebrew translation. The configuration settings shown allow data entry in Hebrew, but user interface text is in English.
7. If you migrate to **ko16ksc5601** from UTF-8, some data may be truncated. You must modify truncated valued because Teamcenter does not support modifying the default field size.

In Hebrew locales, set the following additional variables:

1. In the *TC_DATA***/tc_profilevars** file, set **TC_XML_ENCODING** to **ISO-8859-8**.
2. In two-tier environments, set **TC_CHARACTER_ENCODING_SET** to **ISO8859_8** in the following files:
   - *TC_ROOT***/tccs/Start_TcServer1**

- *TC_ROOT*/**pool_manager/mgrstartMYDB**

Do not set the **TC_XML_ENCODING** or **TC_CHARACTER_ENCODING_SET** environment variables in the system environment. The Deployment Center deploy script, or TEM if you use TEM, sets these values in the Teamcenter configuration.

For non-English locales on Linux systems, you must specify the system locale when logging on to the system using KDE.

B. Special considerations when localizing Teamcenter applications

Audit Manager

If you are localizing Audit Manager, note that the contents of the **Log Handler** list cannot be localized.

Classification

Considerations when localizing Classification

When localizing Classification, there are several points to note:

- You must **install the Classification template** in TEM before you start.
- In Teamcenter, you cannot provide a translation if the master value is not set. But in Classification, you can provide a translation for a display name immediately, before saving. In view mode, you can see the translations but you cannot edit them.
- In the properties of the dictionary, you cannot edit the translations until you add a master value in the dialog box (but you do not have to save it, you simply have to type something).
- You can inherit default class values and overwrite the translations at the class level.
- You cannot enter more characters for a translation than are allowed by the string length in the dictionary.
- With inherited values, you cannot edit or view translations. You must do that at the parent object level.
- All key-LOV, all entries must have approved localized values when you log into the localized rich client before any entry's localized value is displayed. If one of the entries for the localized values is not approved, all key-LOV values display the master value, although some translations may already have an approved status.
- If localization is enabled for Classification, all Classification users can see the **Localization** button. However, Classification users must either have **dba** privileges or be granted specific translation access privileges to enter translations.
- In Classification Admin, you can export translations without translation privileges, but you must have translation privileges to import translations. Similarly, you must have translation privileges to import translations using the **l10n_import_export** utility.

- When exporting translations, the Classification user should always use the transfer modes that begin with **ICSL10N**.

- If you want to work with unit definitions in another locale, you must import the unit definition file corresponding to that locale using the **l10n_import_export** utility. The localized unit definition files are located in the following directory:

  *Teamcenter_root_directory***l10n_cots**

  For example, if you want to see German unit definitions, import using the following command:

  ```
  l10n_import_export -u=user-name -p=password -g=group-name -mode=import
  -file=L10N_classification_unit_definitions.xml
  ```

  Alternatively, you can import the unit definition file using the **Import translated objects** button in the Classification Admin application.

- When localization is enabled, attempting to import a PLM XML file containing key-LOVs that have blank values causes errors in the import and the key-LOVs are not imported. The classification attribute shows question mark symbols in the Classification application in place of the attributes. To correct this, you must edit the PLM XML file manually, adding the missing values, and run the **plmxml_import** utility again to import the updated file. This replaces the question mark symbols in the user interface with valid values.

Installing localization template for Classification

By default, localization is not enabled for Classification. A Business Modeler IDE template is included in the Teamcenter installation kit that you must install using the Teamcenter Environment Manager to enable the localization capabilities for Classification. When you install this template, the system upgrades the Classification schema to support localization capabilities.

Once you install the localization template, the **Localization** button is displayed beside all string values allowing you to enter translations. The following list of properties are automatically designated as localizable:

- Dictionary attribute properties:

  Name

  Short Name

  Default Annotation

  Default Value (**String** format—both metric and nonmetric)

  User 1

User 2

Comments

Help Text

Extended Properties—all the localized **String** properties from Business Modeler IDE.

- Key-LOV properties:

  Name

  Values

- Group properties:

  Name

- Class properties:

  Name

  User Data 1

  User Data 2

- Class attribute properties:

  Annotation

  Default Value (**String** format—both metric and nonmetric)

  User Data 1

  User Data 2

- View properties:

  Name

  User 1

  User 2

- View attribute properties:

User Data 1

User Data 2

Default Value (**String** format—both metric and nonmetric)

Any string attributes present in ICOs are **marked as localizable** at run time.

The Classification localization option appears in the Teamcenter Environment Manager, as described in *Teamcenter Installation on Windows Using TEM* and *Teamcenter Installation on Linux Using TEM*.

Granting translation privileges

By default, only users with **dba** privileges can enter translated values in Classification. To allow users who do not have **dba** privileges to enter translated values in Classification, you must explicitly grant translation privileges in Access Manager. You can:

- Grant privileges to a specific user or group by creating a **Role in Group** entry in the **Translation access control list (ACL)**.
- Grant translation privileges to all Classification objects.

  To do this, add a new rule to the Access Manager rule tree below the **Has Class(POM_object)->in-ClassObjects** node. (This node is located below the **Has Class(POM_object)->System Objects** node in the rule tree.)
- Grant translation privileges for specific types of Classification objects.

  To do this, add a new rule below each object rather than adding the rule below the **Has Class(POM_object)->in-ClassObjects** root node. You can grant access to the following object types:

  - **smlb0**

    Classes
  - **bldb0**

    Views
  - **icm0**

    ICOs
  - **unct_dict**

    Dictionary attributes

- **stxt**

  Key LOVs

Marking classification attributes as localizable

Classification classes and class attributes are created at run time. The classification administrator must mark class attributes of string type as localizable before you can provide localized values for the classification ICO properties.

- To mark a large number of class attributes as localizable, run the **ics_localize_class_attributes** utility, providing the class and attribute ID of each attribute that you want to mark as localizable in the input text file referenced from the **-file** argument.

- To mark individual class attributes as **Localized**, within the Classification Admin application, select **Localized** from the list of attribute properties while adding attributes to a class.

Marking class attributes as nonlocalizable

Teamcenter does not allow the classification administrator to mark a localized attribute as nonlocalizable interactively from the rich client application if the attribute is already referenced by classification ICO objects. Marking an attribute as nonlocalizable requires the deletion of translations of referenced ICO object properties. The classification administrator must log off from Teamcenter and use the **ics_localize_class_attributes** utility with the **-localizable=false** command line option.

Exporting translations for Classification

The **Export translated object** dialog box in **Classification** offers you a subset of options to assist you in exporting string values for translation.

> Caution:
>
> When exporting objects for translation, make sure you click the correct **Export translated object** button  .
>
> Do not confuse this with the PLM XML export button  .

The transfer modes below define the associated objects to be exported for translation along with the selected object. For each object that is exported, all the localizable properties are available for language translation.

Transfer mode	Exports these translations
ICSL10NExportClassHierarchy	Exports all the translations for a selected class with all of its child classes down the hierarchy, and any

Transfer mode	Exports these translations
	associated attribute values, dictionary attributes, and key-LOVs.
ICSL10NExportDefault	Exports the translation string for a single object that is currently selected, for example, a single class or a single ICO and any associated attribute values, dictionary attributes, and key-LOVs. This transfer mode does not include child classes.
ICSL10NExportICO_Admin	Exports the translations for one or more selected ICOs along with translation strings for any associated classification administration object such as view names or dictionary objects.
TIEUnconfiguredExportDefault	Exports the translations for the selected object with no associated data. You can use this transfer mode to export specific objects (for example, those added incrementally after the classification hierarchy is already translated) that helps you to maintain a complete and up-to-date set of language translations.

Note:

Teamcenter exports approved translations only.

There may be situations where the business process dictates bulk translation of multiple objects that are not easy to select in the Classification Admin user interface such as multiple nodes in the classification hierarchy or multiple dictionary attributes. In these situations, the **l10n_import_export** utility may be useful.

Understanding classification localization closure rules

The transfer modes that are delivered with Teamcenter should be sufficient for most translation export situations.

If, however, you need to modify which objects Teamcenter exports for translation, you must modify the transfer modes and closure rules using the following classification object types and related object processing methods.

Related objects (secondary objects)	Represent	Related object process functions	Function description
smlb0	Class or group	**process_ics_class**	Finds and exports related objects of **smlb0** type. Exports associated classes and subclasses.
smlb1	Class attributes	**process_ics_class_attr**	Finds and exports related objects of **smlb1** type. Exports associated class attributes.
bldb0	Views	**process_ics_class_view**	Finds and exports related objects of **bldb0** type. Exports associated views.
bldb1	View attributes	**process_ics_class_view_attr**	Finds and exports related objects of **bldb1** type. Exports associated view attributes.
unct_dict	Dictionary attributes	**process_ics_attr**	Finds and exports related objects of **unct_dict** type. Exports associated dictionary attributes.
stxt	Key-LOVs	**process_ics_attr_stxt**	Finds and exports related objects are of **stxt** type. Exports associated key-LOVs.

For example, the figure shows the use of the classification objects and process functions to configure a typical closure rule for translation export.

Primary O...	Primary ...	Seconda...	Seconda...	Relation ...	Related Property Or O...	Action Type	Conditional C
CLASS	smlb0	CLASS	smlb0	FUNCTION	process_ics_class	PROCESS+TRAVERSE	FUNC_ARGS()
CLASS	smlb0	CLASS	bldb0	FUNCTION	process_ics_class_view	PROCESS+TRAVERSE	FUNC_ARGS()
CLASS	smlb0	CLASS	smlb1	FUNCTION	process_ics_class_attr	PROCESS+TRAVERSE	FUNC_ARGS()
CLASS	smlb1	CLASS	unct_dict	FUNCTION	process_ics_attr	PROCESS+TRAVERSE	FUNC_ARGS()
CLASS	unct_dict	CLASS	stxt	FUNCTION	process_ics_attr_stxt	PROCESS+TRAVERSE	FUNC_ARGS()

ICSL10NExportClassHierarchy transfer mode

Electronic Design Automation (EDA) considerations

Limitations on the ability to localize property values

The limitations arise from the need to store data that is created and consumed by electronic design CAD (ECAD) tools. The following table summarizes these limitations:

Class	Property	Reason
Item	**item_id**	Item ID is a key for looking up designs and components used in designs in Teamcenter. This value must not be localized.
Item Revision	**item_revision_id**	Item revision ID is a key used for looking up specific revisions of a design and must not be localized.
Dataset	**object_name**	Dataset name must not be localized because they may be defined by the ECAD tool.
EDAHasVariant	**variant_name**	Variant name is provided by the ECAD tool and must not be localized.
EDAHasDerivedDataset	**derived_data_name**	Derived data name is a key used to identify the derived dataset created by each derived data configuration. This value must not be localized.
EDAHasDerivedItem	**derived_data_name**	Derived data name is a key used to identify the derived item created by each derived data configuration. This value must not be localized.

Class	Property	Reason
EDADerivedDataConfig	All properties	No property value on this object may be localized.
EDADerivedItemConfigDef	All properties	No property value on this object may be localized.
EDADerviedDatasetConfigDef	All properties	No property value on this object may be localized.

Limitations on locale characters

EDA provides integrations that are designed to work with ECAD tools provided by different vendors. These ECAD tools provide varying levels of support for localization. In order to support the variety of ECAD integrations, EDA imposes the following limitations on the locale characters that maybe used to process ECAD data.

- Partition names must not be localized.
- Only the English versions of all ECAD tools are supported.
- Item ID, revision ID, dataset name, variant name, reference designator value, and mapped attribute names and values must contain US-ASCII characters only. Mapped attributes include CAD attribute mappings and component instance attribute mappings.
- The staging directory and all file names must contain only US-ASCII characters.

Manufacturing

If you are localizing Manufacturing, note the following:

- PAD reports are not localized.
- You cannot populate assets with Business Modeler IDE element names.
- Alias files and publishing templates are not localized.
- ICO mapping in Resource Manager has limitations with multilanguage attributes.
- Guided component search does not support multilanguage scenarios.
- When translating preference names, note that many manufacturing preferences begin with the **ME** prefix. This stands for *Manufacturing Environment*.
- Revision rules are not localized by default and must be **localized**.

Mechatronics Process Management

If you are localizing the ClearCase Integration solution, note that the following ClearCase elements should not be localized:

- **SCMVersionObject**
  - *scm_checkout_view*

    The ClearCase view name used to check out an element.
- **SCMElement**
  - *scm_element_path*

    The path of the ClearCase element.
  - *scm_element_type*

    The type of element.
- **SCMServer**
  - *scm_type*

    For example, ClearCase.
  - *scm_replica_id*
  - *scm_server_uuid*
- **SCMConfigContext**
  - *scm_config_rule*

    The configuration specification string.
  - *scm_config_type*

    The configuration type, for example, label, branch, version.

If you are localizing the Embedded Software Manager solution, note that localization is not supported for:

- Hardware and software compatibility reports.

If you are localizing the Calibration and Configuration Data Management (CCDM) solution, note that the following should not be localized:

- For **ParmDefStr**, the **TableCellStr** involved in minimum value, maximum value, and initial value must not be localized.

Sharing data with other sites using Multi-Site Collaboration or TC XML

If your site participates in a Multi-Site environment or uses TC XML to transfer objects between sites, and the sites support different languages, you must be aware of the following conditions and behavior:

- For any transfer of localized data, the database at both sites must support the same character set encoding.
- Multi-Site environments:
  - Replicated objects are not modifiable at a replica site; therefore, no translations between languages are supported for replica objects.
  - To define the locale for the monolingual site, set the **TC_master_locale_***site-name* preference at your multilingual site that shares data with your monolingual site.
  - When transferring objects between a multilingual and pre-Teamcenter 8.2 site, which is always a monolingual site, attribute values must be in the language supported by the monolingual site.
  - Localization settings at both the exporting and importing site must be consistent. The sites data model representations for localized attributes must be the same.
- TC XML transfers:
  - You must use the **bmide_generatetcplmxmlschema** utility to add the attributes required to support localizable strings to your site's TC XML schema.
  - The **TIE_allow_import_with_different_SML** preference determines if your site allows importing objects in a TC XML file from a site with a different master language.

Localizing Schedule Manager

If you are localizing Schedule Manager, note that the following areas are not localized:

- Schedule Manager reports
- Currency name for costing functionality

- E-mail messages for notifications (sent in same language in which created)
- Interfaces to published service-oriented-architecture (SOA)

  For example, if updating the start date of the task, you are still required to provide the real name, **start_date**, and not a localized UI name such as **Start Date**.

> Note:
>
> Some new screens may require Business Modeler IDE defined names.

Structure Manager

Data that cannot be localized in Structure Manager

If you are localizing Structure Manager, note that the following data cannot be localized:

- Variant options
- Keywords in variant conditions
- Keywords in named variant expressions (NVEs)
- Keywords in revision rule groups
- Keywords in rule checks
- Keywords in module constraints

These limitations apply to the rich client, and any other applications that consume Structure Manager data, for example, Manufacturing Process Planner.

Revision rules are not localized by default and must be localized.

Localize revision rules

By default, revision rules are not localized, and you can deploy English locale revision rules in a localized environment.

1. In Business Modeler IDE, mark the **object_name** property of the **RevisionRule** business object as localizable.
2. Log in to Teamcenter with a user that has Teamcenter administrator privileges.
3. Perform a general search to find all objects of the **RevisionRule** type.

Teamcenter displays a list of all revision rule objects defined in the database.

4. Right-click a revision rule and choose **Edit Properties**.

   Teamcenter displays the **Properties** dialog box.

5. Click the **Localization** button next to the **object_name** property.

   Teamcenter displays the **Localization** dialog box.

6. Enter translations of the selected revision rule in the required locales and click **OK**.

7. Save the properties and check in the changes.

8. Repeat steps 4 through 7 for other revisions rules you want to localize.

9. Start Structure Manager and open the **View/Set Revision Rule** dialog box. Verify that your localized revision rules are available in the list of revision rules.

> Note:
>
> Localized revision rules are available in Manufacturing Process Planner, Multi-Structure Manager, and other applications that use revision rules.

Workflow

If you are localizing Workflow, note that handler names and values cannot be localized.

When exporting Workflow templates containing localized entries, set the **locales** argument of the **plmxml_export** utility to the corresponding locale(s) before export. Export Workflow templates from Workflow Designer by choosing **Tools→Export**.

To localize workflow task names for a workflow process:

1. In the Business Modeler IDE, set the **Localizable** constant to **true** on the **EPMTaskTemplate** business object **template_name** property.

2. In Workflow Designer in the rich client:

   a. Select a process template.

   b. Display the **Properties** dialog.

   c. Provide the localized value for the **template_name** property.

3. Do this for each task in the workflow template.

4. Create the workflow process using the workflow template.

C. Troubleshooting

This appendix describes basic error messages and troubleshooting tips.

- **Two languages appear when deploying Teamcenter**

  After you deploy a localized version of Teamcenter, for example, changing the locale from German (**de_DE**) to English (**en_US**), you may see German text in Teamcenter.

  How to Work Around or Avoid

  Perform the procedure for your specific client:

  - Rich client:

    To prevent mixed-language display the next time you run the rich client after you change the **-nl** argument value, or after you **change your operating system locale**, delete the **Teamcenter** directory under your user directory.

    - On Windows clients, delete:

      C:\Documents and Settings*user-name***Teamcenter**

    - On Linux clients, delete:

      $HOME/Teamcenter[1]

- **Error message appears when entering a localized business object name as a preference value**

  Do not enter localized business object names in preferences. For example, when entering a value in a preference, you must enter the business object name as it appears in the database, not the localized name of the business object.

- **Error message appears when entering a extended character such as Ø.**

  If this error occurs, set the following system environment variable on the Teamcenter host to **TC_POM_MSSQLCOL=latin1_general_bin**.

1 **$HOME** is your user home directory.