



An Oracle White Paper  
December 2010

# Oracle TranslationHub 11g

INTRODUCTION.....	2
Related Documentation.....	2
WHAT CAN BE TRANSLATED .....	2
SUPPORTED LANGUAGES .....	2
GETTING STARTED .....	4
Installing Oracle TranslationHub 11g.....	4
Connecting to a Database.....	4
TranslationHub Database Tables.....	4
CREATING TRANSLATIONS USING TRANSLATIONHUB .....	5
Working with TranslationHub Projects .....	5
Translating Forms Applications.....	7
Using Hubbatch.exe Commands .....	11
DEPLOYING A TRANSLATED FORMS MODULE.....	12
Deploying Multi-language Forms Applications .....	13
WORKING WITH REPORTS APPLICATIONS.....	14

## INTRODUCTION

Oracle TranslationHub is a database oriented translation tool. Using Oracle TranslationHub, you can easily translate the applications built with Oracle Forms and Oracle Reports into any language supported by Oracle products. Separate files can be generated for each language.

### Related Documentation

For information on previous releases of TranslationHub, see:

<http://www.oracle.com/technetwork/developer-tools/forms/documentation/translationhuboverview-129288.pdf>

<http://www.oracle.com/technetwork/developer-tools/forms/documentation/migrate-to-translationhub-132541.pdf>

## WHAT CAN BE TRANSLATED

TranslationHub supports translation of Oracle application files such as .fmb, .mmb, .ogd, .rdf, .res, .msg and .txt. TranslationHub can translate menus, boilerplate text, item labels, messages, and hints defined on item property sheets in Oracle Forms and Oracle Reports applications.

## SUPPORTED LANGUAGES

The following is a list of languages that TranslationHub supports and the default character set for each:

LANGUAGE	DEFAULT CHARACTER SET
American	WE8ISO8859P1
Arabic	AR8MSWIN1256
Bengali	BN8BSCII
Brazilian Portuguese	WE8ISO8859P1
Bulgarian	CL8MSWIN1251
Canadian French	WE8ISO8859P1
Catalan	WE8ISO8859P1
Croatian	EE8MSWIN1250

---

Czech	EE8MSWIN1250
Danish	WE8ISO8859P1
Dutch	WE8ISO8859P1
Egyptian	AR8MSWIN1256
English	WE8ISO8859P1
Estonian	BLT8MSWIN1257
Finnish	WE8ISO8859P1
French	WE8ISO8859P1
German	WE8ISO8859P1
German DIN	WE8ISO8859P1
Greek	EL8MSWIN1253
Hebrew	IW8MSWIN1255
Hungarian	EE8MSWIN1250
Icelandic	WE8ISO8859P1
Italian	WE8ISO8859P1
Japanese	JA16SJIS
Korean	KO16KSC5601
Latin American Spanish	WE8ISO8859P1
Latvian	BLT8MSWIN1257
Lithuanian	BLT8MSWIN1257
Malay	WE8ISO8859P1
Mexican Spanish	WE8ISO8859P1
Norwegian	WE8ISO8859P1
Polish	EE8MSWIN1250
Portuguese	WE8ISO8859P1
Romanian	EE8MSWIN1250
Russian	CL8MSWIN1251
Simplified Chinese	ZHS16CGB231280
Slovak	EE8MSWIN1250
Slovenian	EE8MSWIN1250

---

Spanish	WE8ISO8859P1
Swedish	WE8ISO8859P1
Thai	TH8TISASCII
Traditional Chinese	ZHT16BIG5
Turkish	TR8MSWIN1254
Ukrainian	CL8MSWIN1251
Vietnamese	VN8VN3

## GETTING STARTED

Oracle TranslationHub 11g is packaged as a standalone installation. This section describes the download, installation and initial setup required to begin working with Oracle TranslationHub 11g.

**Note:** Translation Hub is available for Windows platform only. To translate applications on UNIX, copy the files to Windows, translate the applications as given in this document and copy the translated files back to UNIX.

### Installing Oracle TranslationHub 11g

Download Oracle TranslationHub 11g from OTN at:

[http://download.oracle.com/otn/nt/middleware/11g/ofm\\_frm\\_translationhub\\_win\\_11.1.0.0.5\\_32\\_disk1\\_1of1.zip](http://download.oracle.com/otn/nt/middleware/11g/ofm_frm_translationhub_win_11.1.0.0.5_32_disk1_1of1.zip)

Unzip the contents to a folder.

Double click TranslationHub\_11.1.0.0.5.exe.

Follow the instructions to install TranslationHub.

### Connecting to a Database

Use SQL\*Net Easy Configuration Wizard to set up a database host.

Click **Start > Programs > TranslationHub > TranslationHub** to start TranslationHub.

Select **File > Schema Manager** to install, migrate, or delete the schema for TranslationHub.

### TranslationHub Database Tables

The TranslationHub database includes the following tables:

- nx\_project
- nx\_module
- nx\_version

nx\_translation  
nx\_node  
nx\_string  
nx\_attached\_glossary  
nx\_otm\_version  
nx\_access  
nx\_pjt\_seq  
nx\_mdl\_seq  
nx\_vrn\_seq  
nx\_delimiter

## CREATING TRANSLATIONS USING TRANSLATIONHUB

Creating a translation project involves the following phases :

Creating and working with TranslationHub projects

Translating the Applications: You can use TranslationHub to create translations of your software. Optionally, you can choose to update translation projects in the batch mode with `hubbatch.exe`.

Deploying the translated module

### Working with TranslationHub Projects

In Oracle TranslationHub, a project is a logical container for strings. A base translation project contains the original strings. All translated strings are stored in a target translation project. This section describes how to create base and translation projects, and work with strings. For more information on any of these topics, see the TranslationHub Online Help.

### Understanding TranslationHub Views

TranslationHub provides five translation views: the Navigator view, the Attributes view, the Terminology view, the Output view and the Previewer view.

Use the Navigator view to display project data in an Explorer tree, organized by related groups, for example, dialog boxes, menu items, forms, reports, and messages.

Objects selected in the Navigator are displayed in the Attributes View. You can customize the Attributes View to display all of the string data fields for the selected object. Use the Attributes View if you want to edit strings without opening the Item Details dialog or if you want to view more information about each string.

Use the Previewer view to display an approximate representation of the software after translation. This view allows you to see if the translation strings are aligned correctly on screen. The Previewer view displays graphical items if they are attached to the text of selected object.

Use the Output view to display an interactive listing of the output of TranslationHub commands. The Output view consists of tabs. The Standard tab is visible by default. Other tabs appear in the Output view when you run the associated command. For example, the results of running the **Edit > Find** menu item are displayed in the Find/Replace tab. You can save the contents of the Output view in a plain text file.

Use the Terminology view to display base text strings sorted alphabetically. This view groups all identical strings together. It is useful for editing strings that occur more than one and for comparing different translations of the same string.

### Creating a Base Translation

To create a base translation project, select **File > New Project**. Enter the project details. The base language is the language used for development and can be any supported language. You can also choose the database from a list of host names that contain the TranslationHub database schema for storage of the original strings.

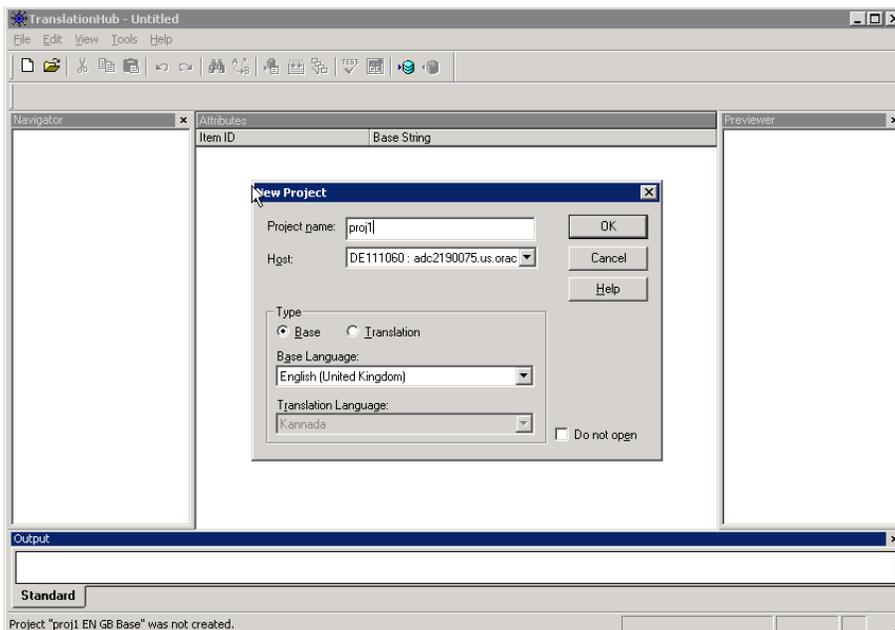


Figure 1: Creating a New Project

### Importing the Base Strings

Once you have created the base translation project, you must import the original strings into the project. Keeping your base translation project open, select **Tools > Import File**. Select the modules

from which you want to import the base strings. You can import strings from Forms Developer modules (.fmb, .mmb files) or Reports developer modules (.rex files). You can further modify the import function in the following ways:

Use Application and Module to group the files, which can later be viewed in the hierarchical navigator view. All application source files will be displayed as branches of the Application name provided. These will be followed by logical modules.

The Move current data to previous data can be used to make a copy of previous strings rather than overwriting them, thus allowing import of the same file several times during development. Both strings are visible for comparison in the Attributes view.

After importing the strings, the Navigator view provides a hierarchical view of the logical application and logical modules. The Preview provides a visual representation of the selected Forms Canvas, while the Attributes view displays all strings in the Base Translation project.

### Merging Translation Projects

Optionally, you can create a project from multiple other projects, or extend a project with strings of another project. Open a project and select **File > Merge Project**. Depending on the type of project (base or translation) you have opened, the Merge Into Project dialog box appears displaying a list of similar projects. Select the project into which you want to merge the current project data. If strings in the two projects are similar, you can either choose to retain existing strings or overwrite strings in the current project. You can also choose to merge data into a remote project by selecting the host details.

### Translating Forms Applications

Once you have created a base translation project with the original strings, there are many different ways you can use TranslationHub to translate your files. The following is an example methodology:

Create the target translation project

Use the Attributes and Terminology views to organize and edit your translation text

Translate text using a glossary

Use Review Comments for reviewing the translated text

Re-use existing translations using the Terminology view

Use Pseudo translate to preview the translation

Build the translated project

Each task in the above process is described in the following sections.

### Creating Target Translation Projects

Target translation projects can be created in two ways:

Create a target translation project in the similar way as explained in creation of base translation projects.

Create a target translation project on the fly before updating it with the strings of the base translation project.

To create a target translation project on the fly, open the base translation project and choose **Tools > Update Translation Project**. The Update Translation Project dialog box displays a list of all target translation projects. You can choose select New Project to create a new target translation project (you can also use Update Translation Project to update existing target translation project with new base strings). Select the languages to create the target translation project.

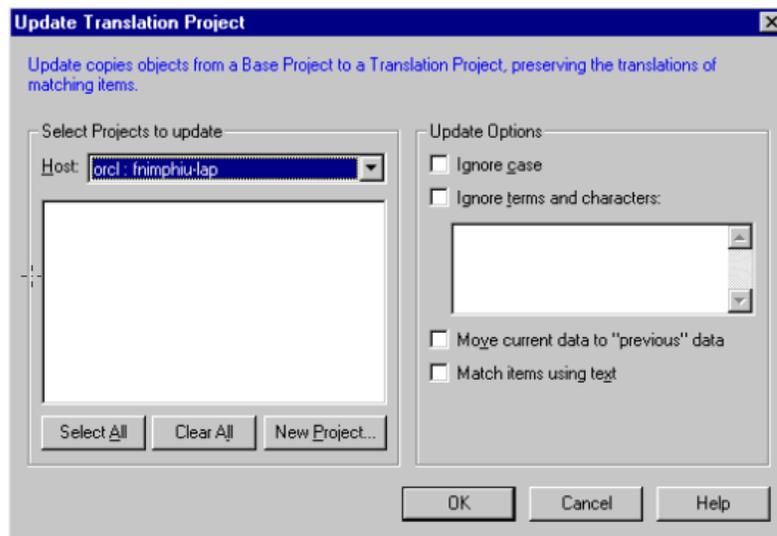
Update options such as the ones listed below will have no affect in the case of a new project:

**Ignore case:** Allows you to update the translation project when there is only a difference of case between base and target translation projects.

**Ignore Terms and Characters:** Enables you to enter terms and characters to be ignored during the update process.

**Move current data to previous data:** Stores a copy of the current translation strings before overwriting them with the imported file content. Both the strings can be viewed and compared in the Attributes view.

**Match items using "text":** Compares the base string of the base translation project with the base string of the target translation project.



**Figure 2: Updating a Project**

Select the options you need and click OK. A target translation project is created with the strings from the base project.

## Using the Attributes and Terminology Views

TranslationHub provides different views to edit strings. The Attributes view is a translation editor that lists all the strings in a tabular style. Strings are displayed in columns that can be customized using the **Choose Columns** dialog. The view can be further customized using color coding (**Edit > Preferences**) to distinguish between base string columns and translation string columns. Color coding can also be used to indicate the translation status of a string. Double click a translation string in the Attributes view. The Item Details dialog is displayed. This window provides information on review status, current translation string, previous string, item's length, position, and alignment on a screen.

The Terminology view is visible only for translated projects and lists the unique occurrences of base strings of the selected module in an Explorer style. Different translations are listed in different folders.

To view the strings of more than one module, select the module in the Navigator view by clicking the left mouse button on the module with the shift or control key pressed. Then open the Terminology view.

If the target translation project includes both Forms and Reports modules, select the root folder to display all the strings in the Terminology view.

Double click a string to view the Terminology Translation dialog. If the selected string is a folder containing child items, the translating the folders base string leads to an update of all the items. As in the Navigator view, the columns in the Terminology view can also be customized.

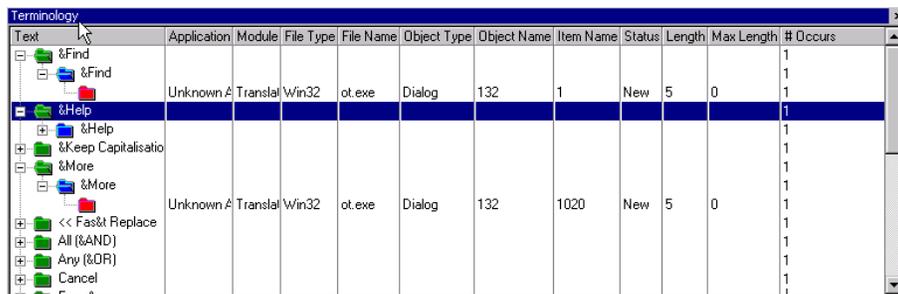


Figure 3: Terminology View

## Translating Text Using the Glossary

Oracle TranslationHub can use glossaries by third-party tools such as Trados MultiTerm to translate a project. To use this feature, install Trados MultiTerm from <http://www.translationzone.com/en/products/sdlmultitermdesktop/> and a Trados MultiTerm glossary for the language you need. Configure the glossary using the Preferences dialog.

## Using Review Comments

Double click a translation string in the Attributes view or Preview view. The Item Details dialog is displayed. For each string, three kinds of comments can be added: a base comment, the translator comment and the reviewer comment. Use these review comments to track reviews of translation strings. A standard review process may involve the following phases:

For any string that may require further research enter a comment in the Translator Comment field. The status of any strings affected is set to Commented.

An editor then reviews the strings and places comments in the Review Comment field. The status of any strings affected is set to Review.

Correct the comments of the editor and manually change the status of any strings affected to Proofread.

When translation work is complete, change the status of the strings to Translated.

### **Re-using Existing Translations**

In large applications that contain several hundred modules, simplify the translation work by selecting a representative number of modules to translate first, import the rest of the modules, and update the translation project.

One of the unique features in TranslationHub, is the ability to view and reuse translations. Select the project root node and open the project in Terminology view. This view displays the base string and groups of any differing translations of the strings. This makes it easy to find different translation of the base string. If translations differ, double click the string and press OK. The view is refreshed with one translation for several occurrences of the base string.

### **Using the Pseudo Translation Feature**

Oracle TranslationHub has a pseudo translation functionality that helps you preview the translated Forms application interface and detect interface translation length even before starting translation. Translation lengths of words may vary because words in different languages could be of different word lengths. For example, the German word “Beruf” is longer than the English word “job” and may require changes in UI positions. A pseudo translation creates random character sequences for each string meeting the average string length of the target translation, and allows you to review the UI.

**Note:** Pseudo translations override the existing strings. Back up the original project before using this feature.

### **Building the Translated File**

A target translation project contains both the base and translated strings. It is necessary to build the translated file to create the translated application. The build process replaces the base strings with the translated strings. To build the translated file, open the translation project and select Tools > Build Translated File. Use the two Browse buttons to select the base application and the folder where you want to store the translated application. You also have the option of selecting files for translation, overwriting files, or selecting files with a specific label.

The translated file is now ready for deployment.

**Note:** For Reports, create the translated version of the .rex file using TranslationHub. Convert the .rex file into an .rdf file or a .rep file using Reports Developer.

## Using Hubbatch.exe Commands

You can create and use a batch file to create projects, update projects, import strings, and build translation files. This feature is useful when you have several files you want to translate or execute similar translations several times during the development process. The following sections describe the commands used, and an example scenario for creating a batch file.

For more information, also see the NOTE:331935.1.

### List of Commands

The following is a list of commands and their syntax for Hubbatch.exe:

Creating a new project: `NewProject,"Project Name","Project Type","Base Language","Base Country",["Trans Language","Trans Country"],"Host Alias",Overwrite`

Opening a project: `OpenProject,"Project Name","Project Type","Base Language","Base Country",["Trans Language","Trans Country"],"Host Alias", Create, Mode`

Merging projects: `MergeProject,"Project Name","Host Alias","Keep Existing Items","File Name"`

Copying projects: `CopyProject,"Project Name","Project Type","Base Language","Base Country",["Trans Language","Trans Country"],"Host Alias","Dest Host Alias"`

Renaming projects: `RenameProject,"Project Name","Project Type","Base Language","Base Country",["Trans Language","Trans Country"],"Host Alias","New Project Name"`

Deleting projects: `DeleteProject,"Project Name","Trans Language","Trans Country","Host Alias"`

Updating projects: `Update,"Project Name","Project Type","Trans Language","Trans Country",Overwrite,ignore Case,update from text,save output`

For updating a project, the Project Type must be "TRANS" and the Base Language and Country must match those of the project that is currently open.

Importing a project: `Import,"Project File","Translation Directory","File Format","Version","Application","Mode",Overwrite,Always Return Success,Save Output`

Build a project: `Build,["base directory"],["output directory"],["overwrite"],["version"],[always return success],[save output]`

Use the parameters "Trans Language","Trans Country" when creating, opening, copying or renaming a translation project. These parameters need not be included for base translation projects.

In the above commands, parameters Create, Mode, Overwrite, Always Return Success, Save Output Build accept boolean values with value 1 for Yes and value 0 for No.

### Creating a Batch file

Create a text file named hubbatchothupd.ini with the following batch commands:

```
[RUN]
```

```
OpenProject, "supermarket","BASE","EN","US","database92",1,0
```

```
Import, "c:\hubbatch\forms_sm.txt","c:\hubbatch\","FORMS10","","Test","SubTest",1,1,1
```

```
Update, "supermarket","DE","DE","database92",1,0,0,1
```

```
OpenProject, "supermarket","TRANS","EN","US","DE","DE","database92",1,0
```

The above commands perform the following:

- Open a project called supermarket, whose base language is English, and is available on a database whose host name is database92

- Import strings into the project

- Update the project, and translate to German

- Open the project with base strings (in this example, English) and translated strings (in this example, German)

Create a batch file named hubb\_othupd.cmd with the following:

```
<directory>hubbatch.exe <location of >hubbatchothupd.ini
```

Start hubb\_othupd.cmd.

You can use Oracle TranslationHub to verify the changes. To build the translation, use the following hubbatch commands to open the project, and build the translated file:

```
OpenProject, "supermarket","TRANS","EN","US","DE","DE","database92",1,0
```

```
Build,"c:\hubbatch\","c:\hubbatch\de\","1","",0,1
```

## DEPLOYING A TRANSLATED FORMS MODULE

To deploy a translated Forms module, you need to first set the NLS\_LANG environment variable. For example, for a German module, set NLS\_LANG to GERMAN\_GERMANY.WE8ISO8859P1. In this environment, compile the translated .fmb file and create the .fmx file. Deploying the translated application involves the following steps:

1. Create an environment file for each translation, for example, german.env from the default environment file, default.env. Specify FORMS\_PATH (indicating the location of menus, and libraries), and PATH (the location of the executables). Additionally, specify the language NLS\_LANG (in this example, GERMAN\_GERMANY.WE8ISO8859P1).
2. Create a configuration section in formsweb.cfg for the translated application, specifying the new environment file (in this example, german.env for an application called my\_application).

```
[my_application]
...
envFile = german.env
...
```

3. Use the URL `http://<host name>.com:8888/forms/frmservlet?config=my_application` to test the application.

For more information on deployment options and how to edit the `formsweb.cfg` and `default.env`, see *Oracle Fusion Middleware Forms Services Deployment Guide* on OTN.

## Deploying Multi-language Forms Applications

The previous section described how to deploy an application translated to one language. In scenarios where a single application needs to be translated to multiple languages, and the same URL is required to access the application, Forms Services makes use of the language encoding in the HTTP header, for example, DE for German, and FR for French. Configuring Forms Services for deploying multi-language applications involves the following steps:

1. Build translated files for each language as described earlier in this document.
2. Set the environment variable for each language and re-compile the translated `.fmb` files.
3. Create environment files, for example, `german.env`, `french.env`, for each language specifying the language environment variable and location of the translated application files.
4. Specify the different environment files in `formsweb.cfg` as follows:

```
[my_application]
...
envFile = default.env
...
[my_application.de]
...
envFile = german.env
...
[my_application.fr]
...
envFile = french.env
...
```

5. Use the URL `http://<host name>.com:8888/forms/frmservlet?config=my_application` to test the application.

**Note:** Custom configuration sections overwrite the default sections. Settings in the `formsweb.cfg` file are accessed in the following order: any language-specific configuration sections (such as `<application_name>.<language>`), followed by configuration sections (`<application_name>`), and then the default System configuration section.

Depending on your browser setting, Forms Services accesses the required environment file. If the browser setting has German (DE), Forms Services looks for an entry such as `<application name>.de`. It then locates the environment file, in this case, `german.de`, which in turn specifies the location of the translated application files.

If the browser setting has French (FR), Forms Services looks for an entry such as `<application name>.fr`. It then locates the environment file, in this case, `french.env`, which in turn specifies the location of the translated application files.

If no language definition is found, then the default `[my_application]` entry is used to start the application.

Thus, by using the same URL, different users can access different translations of the same Forms applications.

## WORKING WITH REPORTS APPLICATIONS

Translating Reports applications is similar to translating Forms applications. To call embedded Reports in a translated Forms application, set the Reports Server with the environment of the language used. `Reports_Path` must point to the location of the translated version of the Reports module.



Oracle TranslationHub 11g  
December 2010  
Author: Grant Ronald, Hendrik Schmidt,  
Contributing Authors: Vandana Rao

Oracle Corporation  
World Headquarters  
500 Oracle Parkway  
Redwood Shores, CA 94065  
U.S.A.

Worldwide Inquiries:  
Phone: +1.650.506.7000  
Fax: +1.650.506.7200

oracle.com



Copyright © 2010, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. UNIX is a registered trademark licensed through X/Open Company, Ltd. 1010

**Hardware and Software, Engineered to Work Together**