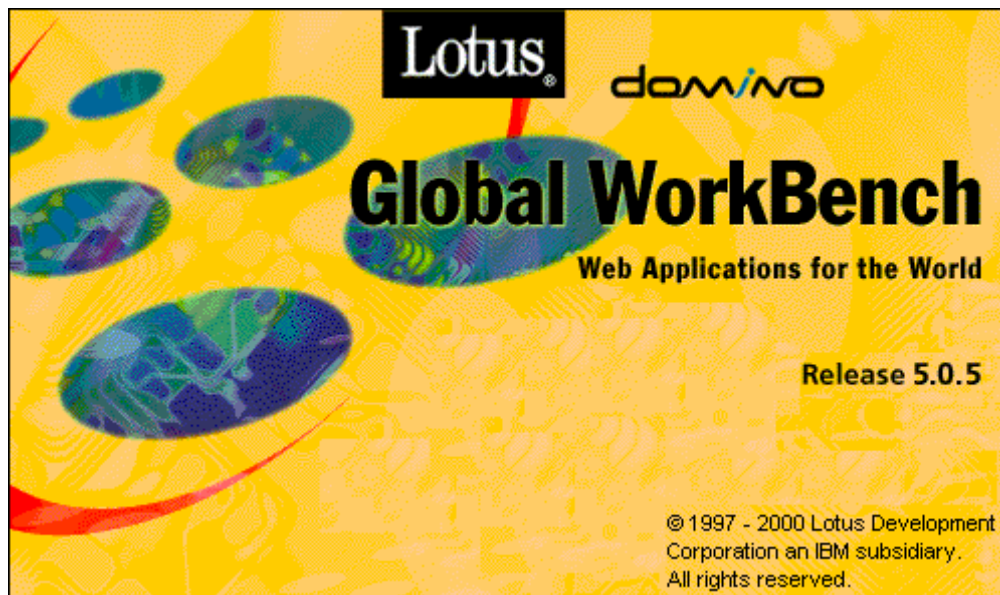

Domino Global WorkBench

Guidelines for Developers and Translators



Copyright and trademark information

Disclaimer

THIS DOCUMENTATION IS PROVIDED FOR REFERENCE PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS DOCUMENTATION, THIS DOCUMENTATION IS PROVIDED "AS IS" WITHOUT ANY WARRANTY WHATSOEVER AND TO THE MAXIMUM EXTENT PERMITTED, LOTUS DISCLAIMS ALL IMPLIED WARRANTIES, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE, WITH RESPECT TO THE SAME. LOTUS SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION, DIRECT, INDIRECT, CONSEQUENTIAL OR INCIDENTAL DAMAGES, ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS DOCUMENTATION OR ANY OTHER DOCUMENTATION. NOTWITHSTANDING ANYTHING TO THE CONTRARY, NOTHING CONTAINED IN THIS DOCUMENTATION OR ANY OTHER DOCUMENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM LOTUS (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF THIS SOFTWARE.

Copyright

Under the copyright laws, neither the documentation nor the software may be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form, in whole or in part, without the prior written consent of Lotus Development Corporation, except in the manner described in the documentation or the applicable licensing agreement governing the use of the software.

© Copyright 1985 - 2000 Lotus Development Corporation
55 Cambridge Parkway
Cambridge, MA 02142

All Rights Reserved. Published in the United States.

List of Trademarks

Domino, cc:Mail, Notes, Domino Global WorkBench and Notes/FX are trademarks and Freelance, Freelance Graphics, Lotus, Lotus Notes, LotusScript, NotesSQL, NotesView, 1-2-3, Organizer, SmartIcons, and SmartSuite are registered trademarks of Lotus Development Corporation. OS/2 Warp and PowerPC are trademarks and AIX, IBM, OS/2, Presentation Manager and SNA are registered trademarks of International Business Machines Corporation. Adobe and Adobe Acrobat are registered trademarks of Adobe Systems Incorporated.

All other trademarks are the property of their respective owners.

Table of Contents

Chapter 1

Introduction to Domino Global

WorkBench 1

Overview of Domino Global WorkBench 1

 The WorkBench 3

 About Domino Global WorkBench databases 4

The overall Domino Global WorkBench process 7

Who does what? 10

 Localization analysis 11

 Tagging and test building 11

 Translation preparation 11

 Translation 12

 Building and checking 12

 Updating 12

 Quality assurance 12

What's new in Release 5.0.5 of Domino Global WorkBench? 12

Upgrading existing Domino Global WorkBench projects 13

Upgrading Notes Global Designer Release 4.6 projects 14

Using Domino Global WorkBench with Notes Release 4.6 databases 15

Support for bi-directional languages 17

Setting up 18

 System requirements 18

 Installation 19

 Documentation 21

Chapter 2

Recommended processes

Localizing Domino databases 23

 Step 1: Prepare the source database 23

 Step 2: Create a glossary 23

 Step 3: Start the WorkBench 24

 Step 4: Create the tagged database 24

 Step 5: Review the glossary 25

 Step 6: Create 'pseudo' translations 25

 Step 7: Build a pseudo-language database 26

 Step 8: Test the built database 26

 Step 9: Translate 27

 Step 10: Build language database(s) 27

 Step 11: Update 27

Creating synchronized databases 28

 Step 1: Select the language databases 28

 Step 2: Select the synchronization type 29

 Step 3: Set user options 29

 Step 4: Add the synchronization features 29

 Step 5: Deploy the database(s) 29

Adding translation workflow 30

 Step 1: Preliminaries 30

 Step 2: Create the workflow control database 30

 Step 3: Assign roles and access rights 31

 Step 4: Create the workflow information document 31

 Step 5: Edit the language information documents 32

 Step 6: Schedule the workflow agent 32

Chapter 3

Preparing source databases for localization

International considerations 33

 Acronyms 33

 Colors 33

 Culture 34

 Expansion 34

 Fonts 34

 Formats 35

 Graphics 36

Good programming practice 36

 Using aliases 37

 Avoiding concatenated sentences 39

 Keywords 39

 Formulas and script 40

 Handling text that never displays 42

 HTML 44

 Java 45

 Using existing terminology 46

 Keeping translators informed 46

Preparing a database for building multilingually 47

Useful @functions 48

Limitations 49

 Attachments and embedded items 49

 Design elements marked 'Prohibit refresh or replace' 49

 Help documents 49

 Length limitations 50

 Referencing other design elements 50

 Simple actions 51

 Subforms 52

Modifying source databases after localization has started 52

Preparing an existing database 53

 Making a backup 53

 Creating a design synopsis 53

 Reviewing a design synopsis 55

 Creating or changing aliases 56

Chapter 4

Working with glossaries 59

Local, Project and Global glossaries	61
Term definition documents	61
Translation documents	62
Reference language	63
Planning	63
Creating a glossary	64
Making a language active	67
Making a language inactive	69
Changing the reference language	70
Term types	70
Reviewing a glossary before translation	71
Managing a glossary	72
Upgrading a glossary from Notes Global Designer 4.6	72
Roles	72
Adding categorization to glossary views	73
Creating new terms	73
Importing, exporting and retrieving terms and translations	77
Deleting terms	80
Locating and marking obsolete terms	81
Housekeeping tasks	81
Creating 'pseudo-translations'	82
Replicating a glossary	84

Chapter 5

Translating 87

Translation preliminaries	87
Commenting glossary terms for translators	88
Terms that must not (or need not) be translated	89
Text that should be translated	92
Locating text to exclude from translation	92
Preventing the translation of terms in a glossary	93
Translation views	94
How to translate	96
Using the Translate/Review dialog	96
Translating a translation document directly	98
Duplicate glossary entries	100

Chapter 6

Using the WorkBench 101

The WorkBench display	101
Projects	102
Creating a project	102
Opening an existing project	103
Loading source databases	103
Loading glossaries	104
Creating a tagged database	106
Building language databases	113
Defining language databases	114
Selecting design elements for building	115
Creating (building) language databases	116
Batch processing	118
Updating	119

Rebuilding databases	123
Tagging	124
Tags	125
Tagging in general	125
Tagging specific design elements	129
Tagging limitations	141
Items the WorkBench does not tag	142

Chapter 7

Checking and using built databases 143

Functionality in language databases	143
Design changes in language databases	144
Switching languages in built databases	146

Chapter 8

Language synchronization and translation workflow 149

Language synchronization	150
Implementing language synchronization	152
Preparing a database for language synchronization	152
Fields in synchronized databases	154
Installing the Language Synchronizer and the Synchronizer configuration database	156
Adding language synchronization features	156
Using the Synchronizer configuration database	161
Using synchronized databases	163
Translation workflow	166
Preparing a database for translation workflow	168
Setting up translation workflow	169
Using translation workflow	174
Canceling translation workflow and deleting workflow documents	177
How translation workflow works	178
Replicating synchronized and translation workflow databases	185

Appendix

Error messages 187

WorkBench error and warning messages	187
Glossary error and warning messages	205
Error messages	205
Warning messages	212
Synchronization error and warning messages	214
Translation workflow error messages	217

Index 221

Chapter 1

Introduction to Domino Global WorkBench

Domino Global WorkBench™ is a set of software tools that helps you to manage the localization (translation) of Domino databases, especially Web site databases.

The WorkBench automatically extracts terminology from your database application and stores it in glossaries ready for translation. Localized versions of the database are built automatically using the translated glossaries. And if your database changes, you can use the WorkBench's update features to transmit the changes easily through to the localized versions. Domino Global WorkBench can also add translation workflow features to built databases. These help you to manage the localization of documents within your built databases.

Localization

For a database (.NSF file) or database template (.NTF file) 'localizing' means creating a version of it in another language. In general, localization involves:

1. Analyzing and sometimes modifying the design to make the remaining localization tasks simpler.
2. Identifying text that must not be translated (for reasons of functionality) or need not be translated (because it is not seen by users).
3. Translating the remaining text.
4. Modifying the layout. Translations are often significantly longer than the original text, and this means that certain parts of the design, for example tables and navigators, will usually have to be adjusted.

For more information see "Who does what?", below.

Note Domino Global WorkBench is primarily concerned with translating the *design* of a database. Translating the *content* of a database (in other words the data documents it contains) may also be desirable, especially in a Web site database, and so Domino Global WorkBench includes language synchronization features to help you with this.

Overview of Domino Global WorkBench

A Domino Global WorkBench 'project' comprises:

- The **database(s)** you want to localize. These might be new databases in the course of development, or existing databases now destined for use internationally. They are the 'source' databases.

Domino Global WorkBench is primarily intended for localizing the design of a database, rather than the data documents it contains. Documents that are part of the design, such as small Help documents, can be localized using Domino Global WorkBench. Larger documents are best localized separately.

- One or more **glossaries**. These are Notes databases, built on a standard template, that hold terminology from the source database(s). Each entry in the glossary has one or more translation documents associated with it. Translators translate the text in the translation documents.
- For each source database a **tagged database**, which provides a link between the text in the source database(s) and the glossary entries.
- For each source database, one or more built **language databases**. These are the localized databases that people will use. Typically they comprise a multilingual Web site. There are two types of language database:

A standard database (or template), with its design elements localized in one language. This is termed a 'unilingual' database.

A multilingual database (or template), with its design elements localized in several languages. Users can switch from one language to another from Notes or from their Web browser.

Language synchronization features can be added to the language databases to support translation workflow. See Chapter 8, "Language synchronization and translation workflow".

- Entries in the project database (PROJECT.NSF, in the Domino Global WorkBench program directory), which stores information about Domino Global WorkBench projects.
- (Optional) Records of WorkBench operations stored in a reports database.

Some advantages of using Domino Global WorkBench

Domino Global WorkBench provides a number of advantages over conventional localization methods, including:

Re-use of technical up-front investment One technical person can do the initial analysis of the application, define what needs to be translated and what should be left in the source language, provide contextual comments for translators, and so on. This effort speeds the work of *all* translators, who do not have to concern themselves with, for example, what should and what should not be translated (in fact much text that should not be translated they do not even see).

Efficient use of resources Technical people perform technical tasks, whereas translators concentrate on translation.

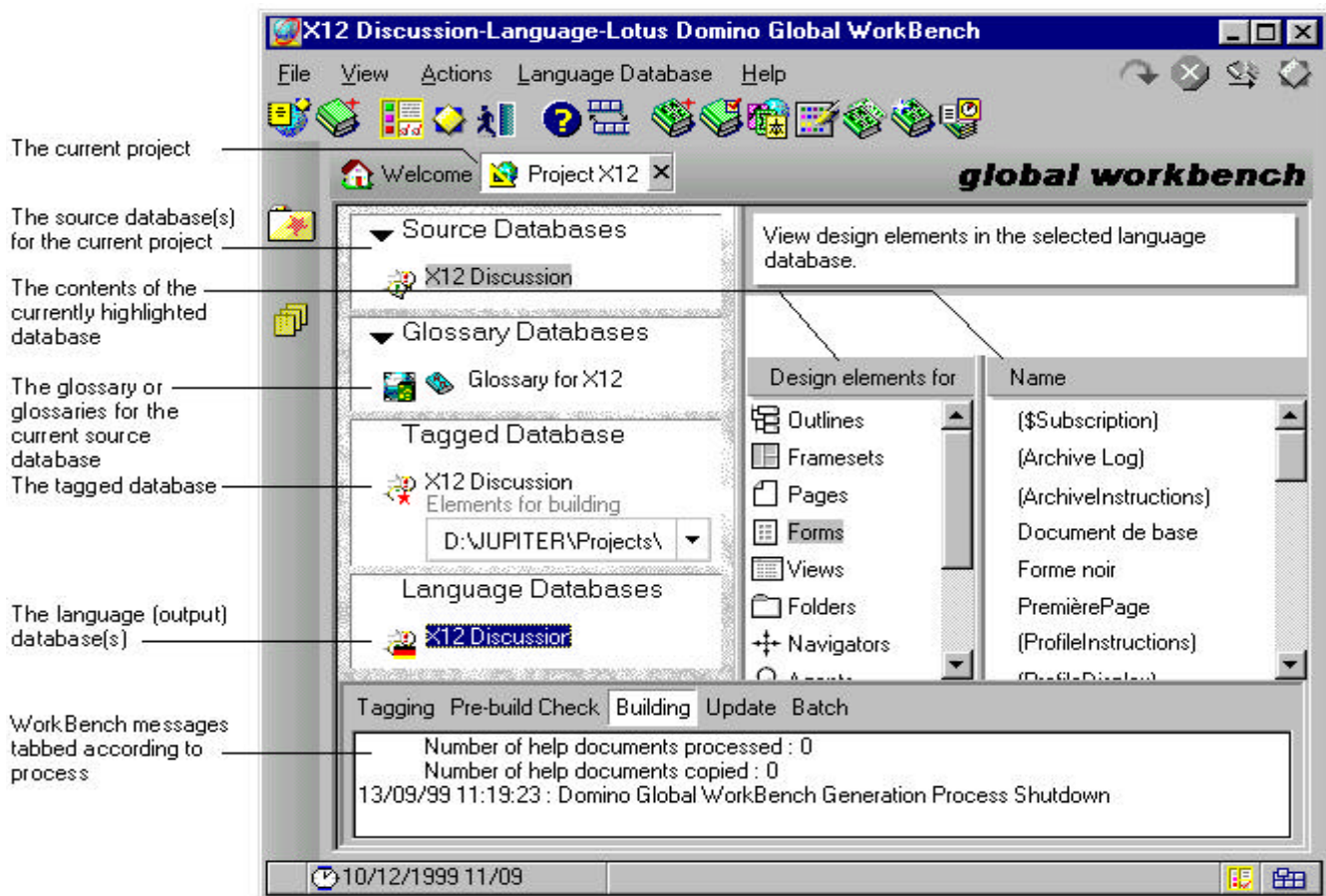
Translation is done with full contextual information Translators know at all times what design element they are translating and they can always trace it back in the source application.

Re-usability of translations Your translation investment is protected. Because all translations are stored in glossary databases with full contextual information, they can be re-used in other applications that use similar terminology.

Multilingual applications You can build several translated databases from a single source, or you can build one multilingual database with design elements in several languages. This simplifies the creation of multilingual Web sites.

The WorkBench

The **WorkBench** is a Windows (95, 98, NT or 2000) application which provides all the key Domino Global WorkBench functions. It uses 'Click Here' prompts to guide you logically through the Domino Global WorkBench process (creating a project, selecting a source, loading a glossary, creating a tagged database, and finally building language databases or localized resource files). The WorkBench displays all the files that make up the current project, and it uses a system of color coding to show differences that might necessitate an update.



The WorkBench does the following:

- Creates tagged databases from their respective source databases. The WorkBench does this automatically. Each piece of text or bitmap in the source database is looked up in one or more loaded glossaries. If a match is found, the text or bitmap is tagged. If a match is not found, you have the option of adding the text or bitmap to one of the glossaries and then tagging to it.
- Builds localized language databases (these are the databases users will see and use) from the tagged database and its associated glossaries. You can choose to create:
 - Database(s) which contain design text in several languages. In such databases the database users decide which language they want to see displayed on the screen.
 - Database(s) which contain design text in only one language.

You can choose to add synchronizing features to either type of database.

The WorkBench can create databases in .NTF or .NSF format, and can create them either as copies or replicas of the source. The output databases are Domino-compliant.

A report produced by the WorkBench identifies any problems that may arise during creation. For example, the report will identify any pieces of text that lack translations in the desired output language(s). Translators can then add the missing translations to the glossary using Notes.

- Updates the tagged database, the glossaries, and the language databases if the design of the source database changes.
- Logs process messages to a log window, and, optionally, to a reports database.

The WorkBench stores project information in the project database, PROJECT.NSF.

See Chapter 6, "Using the WorkBench".

About Domino Global WorkBench databases

You will encounter the following types of Domino database while using Domino Global WorkBench:

- Source databases
- Glossaries
- Tagged databases
- Language databases
- Multilingual databases
- Synchronized databases

Source databases

These are the database(s) you are localizing. They might be new databases in the course of development, or existing databases now destined for use internationally.

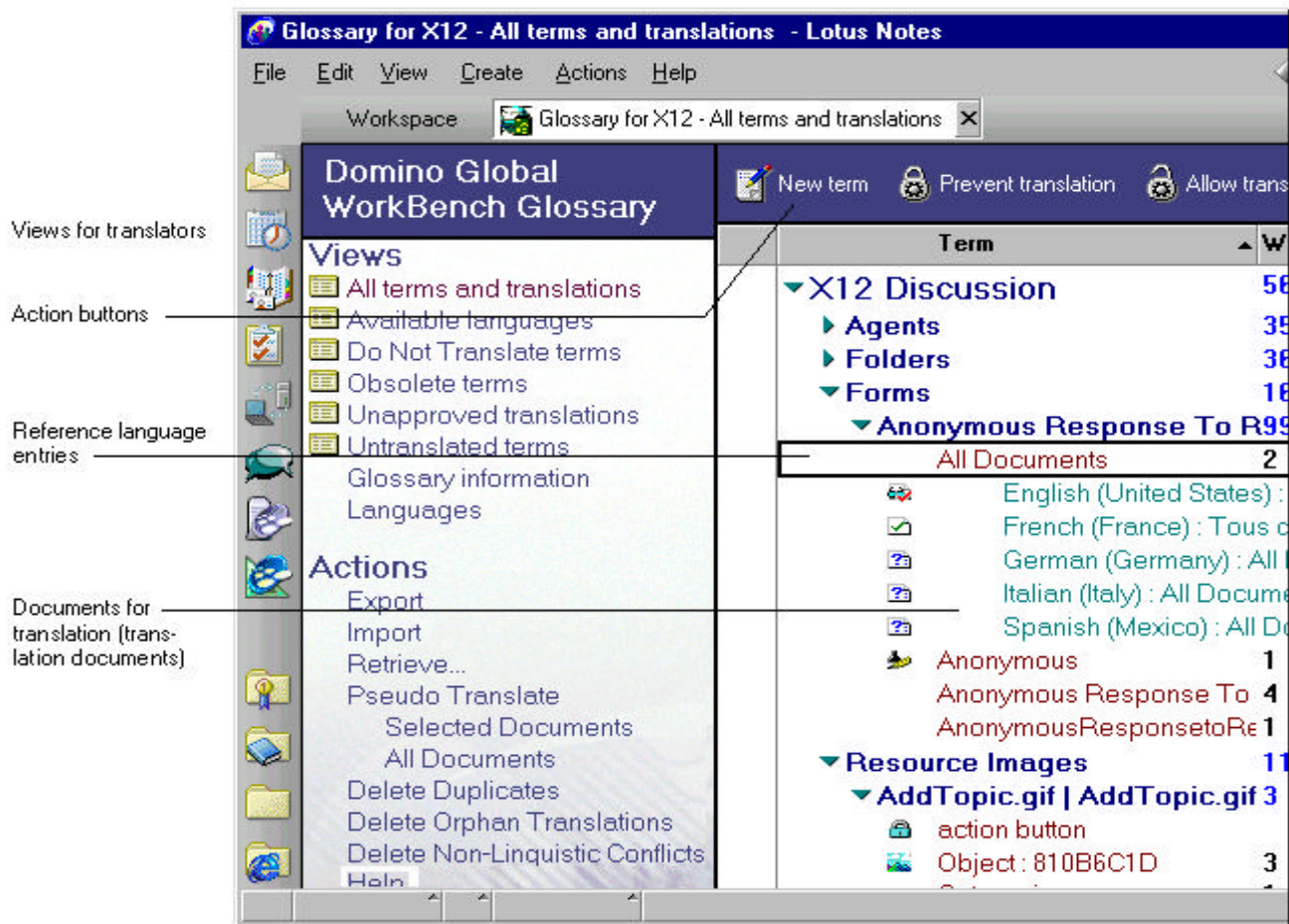
Domino Global WorkBench is primarily intended for localizing the design of a database, rather than the data documents it contains. For this reason a source database is usually a template (NTF) file, though it can be an NSF file. Documents that are part of the design, such as small Help documents, can be localized using Domino Global WorkBench. Larger documents are best localized separately.

Glossaries

A Domino Global WorkBench glossary is a Domino database. It is created in the usual way (File - Database - New) using the template DGWGLOSS.NTF. Once you have created the glossary database and specified the languages you want it to support, you enter your terminology into it. You can do this in various ways:

- By using the WorkBench. Glossary entries are created automatically when the WorkBench creates a tagged database (see below).
- By adding new terms or translations in Notes.
- By importing terms from a structured text file using Notes.

The following illustration shows a Domino Global WorkBench glossary database containing terminology.



The glossary contains:

- A term definition document for each term defined.
- For each term definition document, one or more associated translation documents. These are the documents that translators translate. Each translation document contains the text of the term for one language. Icons indicate whether or not a translation document has been edited (translated).

Terms in the glossary are categorized by context, so, for example, all the terms that come from forms are displayed together under the appropriate form names. The WorkBench does this categorizing automatically when it creates glossary entries.

Some text in a database application must not be translated (aliases referenced in formulas, for example). You can prevent terms from being translated by selecting them in the glossary and clicking the Prevent Translation button. An icon indicates when a term has been marked in this way.

Because glossaries are Domino databases they can easily be replicated to any Notes site that needs them, for example translation sites.

Tagged databases

Tagged databases provide the essential links between text in a source database and entries in glossaries. A tagged database is a replica of the source database and identical to it, except that most of its text is 'tagged', linking it to glossary entries. A tag is a unique ID, which the WorkBench attaches to the beginning of each piece of text. The WorkBench further encloses each tag-plus-text combination within special delimiters (the defaults are {~ and ~}) so that it can distinguish it from untagged text. You can change these delimiters if your source database contains these character combinations.

You can see tagged text by highlighting a tagged database in the WorkBench display.

Tagged databases are for use by the WorkBench and you should not generally need to touch them. If you want to modify your database in any way, you should make changes in the source database. The WorkBench update process will transfer these changes to the tagged database, and from there to the language database(s).

Note The open and close brace characters, { and }, are used as literal string delimiters by LotusScript. If you use these characters in tag delimiters for the WorkBench, you may see some anomalies when you view the tagged database. However, this will not cause problems in the built language databases.

Language databases

The databases built by Domino Global WorkBench are called 'language databases'; they are the translated (localized) databases that people will use. For each source database you can build one or more language databases, and these can be:

- Multilingual databases which contain design elements in several languages. This allows, say, French users to see the user interface in French, while German users see it in German, and Japanese users see it in Japanese. This is useful for databases that are to be accessed by people speaking different languages in the same (or any) site. Users can switch the display of the database design from one language to another from Notes or from a Web browser.

See "Switching languages in built databases" in Chapter 7.

- Unilingual databases, which are standard Domino databases, each with a different design language. This is useful for databases that are to be accessed by people speaking different languages, but where access is restricted to particular replicas (for example, French speakers access a replica in Paris, English speakers access a replica in Boston, Japanese speakers access a replica in Tokyo, and so on).

Language databases can be built in .NSF or .NTF format, and can be either a copy or a replica of the source database. They can be synchronized or not (see "Synchronized databases", below).

Multilingual databases

One of the features of Domino Global WorkBench is its ability to build 'multilingual' databases. In a multilingual database, the design elements are present in several different languages. Each user sees the user interface, in whatever language he or she prefers and can switch from one language to another by changing the language setting in Notes, or in their Web browser. See "Switching languages in built databases" in Chapter 7.

Note An exception to this is the database title and the database icon. In a multilingual database these will always be the default language ones: they do not change according to a user's language settings.

You can use this feature to create a multi-language Web site in a single database, rather than having a separate database for each language. To build a multilingual database you simply select more than one language on the WorkBench's New Language Database dialog box when you build.

When you build a multilingual database the WorkBench automatically builds out each design element in each of the languages you selected, adding to each a standard language identifier (for example de-DE for German). Then, if a user specifies his or her language as German (in the Notes client or Web browser), s/he will see the de-DE version of the form when creating a document. This is handled automatically by the Domino Release 5 server for Web users, or the Notes Release 5 client for Notes users.

When designing databases that will be built as multilingual databases, there are a few things that you will need to pay attention to. See "Preparing databases for building multilingually" in Chapter 3.

Synchronized databases

You can add language synchronization features to any language database (multilingual or unilingual) to create a synchronized database. This allows *data documents* to be synchronized across languages. In a synchronized database users can mark their documents in one of three ways:

1. **Translatable.** In the language database(s), any document created from this form will be copied automatically to other languages and marked for translation.
2. **Global.** In the language database(s), any document created from this form will be copied automatically to other languages but will not be marked for translation.
3. **Local.** In the language database(s), any document created from this form will not be copied to other languages.

The Language Synchronizer, a scheduled server task, does the necessary copying work. It also handles changes to existing synchronized documents.

You can synchronize between unilingual databases or within multilingual databases. If you are using unilingual databases, you can also choose to build a Language SwitchBar into synchronized documents. This allows users to navigate to other language versions of any synchronized document.

Language synchronization provides the basis for translation workflow, an automated process for handling the translation of documents in built databases.

See Chapter 8, "Language synchronization and translation workflow".

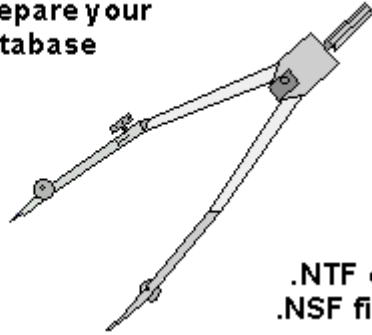
The overall Domino Global WorkBench process

There are seven stages in the process, the last two being optional. These are outlined below, and each is explained in more detail in the following chapters.

Note See Chapter 2 for more details of Domino Global WorkBench processes.

1

Prepare your database



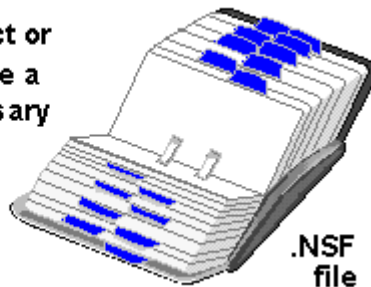
**.NTF or
.NSF file**

You start with the Domino database that you want to localize (translate). This is the 'source' database. It will usually be a template (NTF) file, but it can be a regular database (NSF) file. There are some useful preparatory steps that you can take to simplify the Domino Global WorkBench process, for example by creating aliases.

See **Chapter 3**.

2

Select or create a glossary



.NSF file

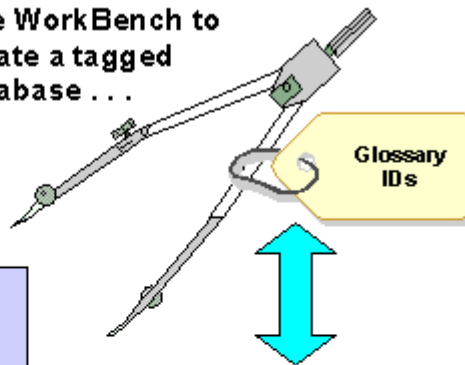
You create a glossary. This is a Domino database (NSF file) based on the template DGWGLOSS.NTF. You specify what languages you want the glossary to support.

To begin with, the glossary will be empty. You can import existing terminology if you wish. You can also use more than one glossary at once, and you can use existing glossaries if you have them.

See **Chapter 4**.

3

Use WorkBench to create a tagged database ...



... linked (by tags) to ...

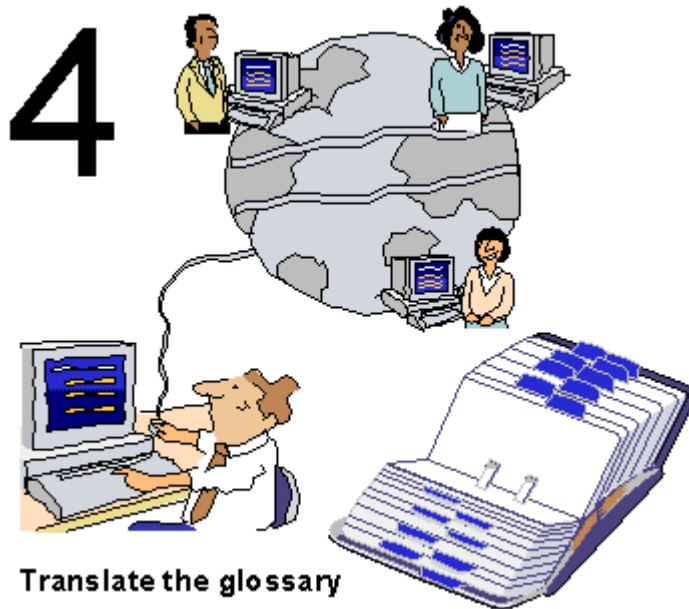


... the glossary

Use the WorkBench to create a tagged database from the source database. This is identical to the source database, but each piece of translatable text in it is given a unique ID, a 'tag', that links it to an entry in the glossary (or glossaries). As it creates the tagged database, WorkBench looks up each piece of text in the chosen glossary or glossaries, creating tags to existing glossary entries whenever it can. If it finds no match it creates a new glossary entry and tags to that.

It is important to know what to translate and what not to translate before you start using WorkBench.

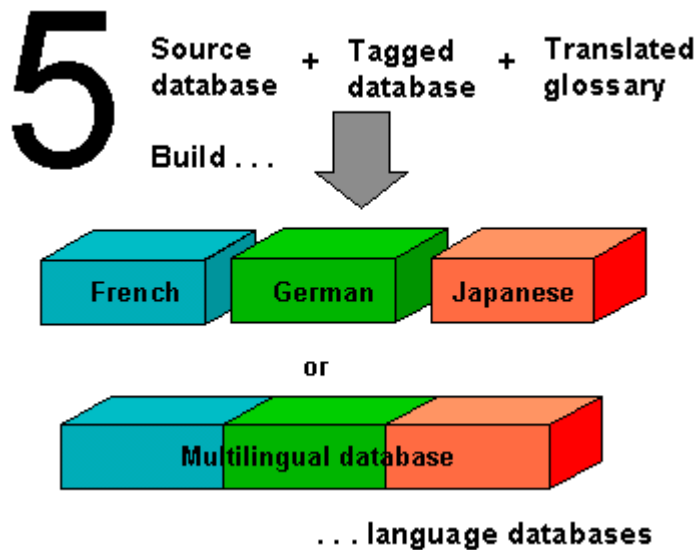
See **Chapter 6**.



Translate the entries in the glossary using Notes. The glossary file is a regular Domino database (.NSF file) and can therefore be replicated to translation sites worldwide.

There is also a 'pseudo-translation' function, which allows you to see the effect of changing text in your application before any real translation is done.

See Chapter 5.



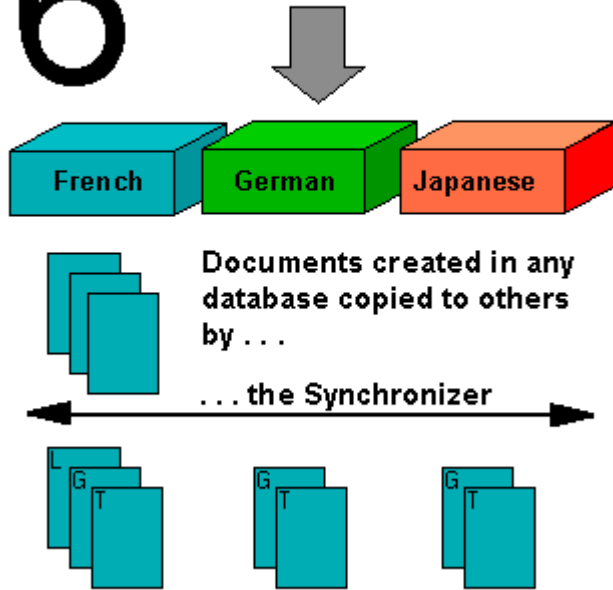
Build language databases from the source and tagged databases and the translated glossary. Language databases are the databases that people will use.

The language databases can be .NSF or .NTF format. And language databases can be multilingual: the user decides what language s/he wants to see displayed.

Domino Global WorkBench lets you update your built databases (and your glossaries) if the source database changes, without recreating them from scratch.

See Chapter 6.

6 Add synchronization features

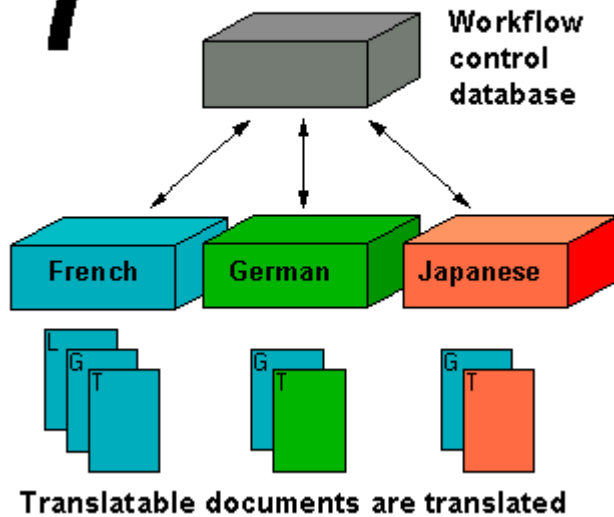


Optionally, add language synchronization features to the built language databases. This allows documents created in them to be marked as 'local' (L), 'translatable' (T) or 'global' (G).

The Language Synchronizer, a server task, copies translatable and global documents created in any of the databases to the other databases in the synchronized set.

See Chapter 8.

7 Translation workflow



Optionally, set up translation workflow in the synchronized databases.

This is a Domino application that guides translatable documents through a translation process, which you define.

The relevant people are notified by mail as each step is completed until the translations finally become 'approved' and are made visible to database users.

See Chapter 8.

Who does what?

There are a number of tasks in the localization process, and they require people with different skills. The following sections provide some advice about the skills and knowledge different contributors should have.

Localization analysis

This is a job for an experienced engineer and a translation specialist. Ideally the engineer will be one of the developers of the application, or at least have good lines of communication with the developers. The translation specialist should be an experienced translator with a good knowledge of Notes and Domino.

The following tasks should be carried out, and any problems reported to the developers of the application:

- Check that the application supports the display of fonts and characters needed by the target languages (for example accented characters and Asian character sets).
- Check that the application is ready for processing by Domino Global WorkBench, in particular that aliases with suitable identifying prefixes have been used. Although not strictly speaking essential this will save a lot of work later.
- Check translatability. Look for potential translation problems such as impossible concatenations, use of graphics and text, splitting of sentences, buttons with text that are too small to allow translation, table columns that are too narrow, the use of internationally unsuitable fonts, and so on.

See Chapter 3, "Preparing source databases".

Tagging and test building

Again, this is a job for an engineer, ideally one of the developers of the application. The following WorkBench tasks should be carried out:

- Tag the application. This connects the application with a glossary, making new glossary entries where necessary.
- Mark glossary entries that must not be translated (called DNTs, 'do not translate'). Text that is never seen by users, or which for functionality reasons must remain in the source language, must be marked 'do not translate'.
- Run test builds using a pseudo-language. A pseudo-language is a jumbled and expanded version of the source language. It allows the effects of translation to be examined before any real translation is done.
- Check the test builds and report problems that arise from the design of the source to the appropriate developers.

See Chapter 4 "Creating and working with glossaries" and Chapter 6 "Using the WorkBench".

Translation preparation

A translation specialist should review the contents of the glossaries, especially the DNT information, and provide translation comments to give further context to translators (for example, explain how formulae and concatenations work, indicate whether a piece of text is a box title, a button label or whatever). The translation specialists should look at the glossaries from a translator's standpoint and try to make his or her work as simple as possible.

See "Translation preliminaries" in Chapter 5.

Translation

Translators translate the entries in the glossary. If the previous tasks have been carried out, their task will be minimized. The translators should be experienced translators of computer software and they should be conversant with Notes and its terminology.

Building and checking

Once the build process has been perfected by test building (above), building the real language versions of the application should be straightforward. It will however be necessary for the translators to check that no source language text remains visible in the built versions.

Updating

In an ideal situation you will be localizing a finished database application. However, it is likely that changes will be made at some stage to the original source database. Preparation of updates is a job for the engineer who did the tagging and test building (above). The translators will have to translate any new text, and the language databases rebuilt and rechecked.

Quality assurance

Finally, the localized databases should be tested. A quality assurance engineer should have a sound technical knowledge of Notes and standard Web browsers and versions. It will help if a test script is developed early in the localization process which can then be applied to all built databases.

What's new in Release 5.0.5 of Domino Global WorkBench?

Domino Global WorkBench Release 5.0.5 differs from earlier releases in the following major respects:

- Translation workflow provides you with an automated process for handling the translation of documents in built databases, typically Web site databases. Translators and reviewers are notified automatically whenever new translatable documents are created or existing ones are modified. After passing through various steps of a translation process, which you specify, documents are finally marked as approved. Only then do they become visible to users.
- Synchronization (now called 'language synchronization') has been enhanced. The Synchronizer (renamed the 'Language Synchronizer') is now a server task, scheduled from a central database, the Synchronizer configuration database, and language synchronization features are added to databases from a separate WorkBench display. Synchronized unilingual databases no longer have to be put into the same directory, and they can be located on different servers. The Synchronizer report database is no longer used.
- Text in JavaScript, whether native or in pass-thru HTML, is tagged for translation.
- The localization of Java .properties files is now supported.
- The WorkBench now supports Unicode. If you have installed Unicode fonts, the WorkBench will automatically use one that is appropriate for the language it is displaying.
- Updating now relies on UNIDs rather than NoteIDs. This means that updating continues to work when the source database is replaced by a copy or replica of itself.

- The localization of profile documents is now supported. They are treated as Help documents.
- The display of Help documents in the WorkBench has been enhanced.
- A Create button on the Select Glossary dialog lets you create a new glossary database directly from the WorkBench.
- A 'Set Settings as Default' button on the tagging and build dialogs allows you to make the current settings the defaults for new projects.
- A new option on the Tagging Options dialog allows you to mark glossary terms (with a piece of text you specify) during tagging. This helps you to identify glossary terms that are no longer being used.
- Two new options have been included on the Build Options dialog. 'Replace fonts' lets you use alternative fonts when building databases. 'Preserve layout' lets you retain format changes when updating databases.
- In the glossary, buttons in translation documents and the Translation and Review dialog let you apply translations automatically to other translation documents with the same parent text. The pseudo-translation function has been enhanced.
- The project manager is no longer a part of Domino Global WorkBench. Instead, icons on the left of the WorkBench display allow you to bookmark projects and organize them in folders. Right-clicking displays a menu of project options, such as Create, Open, Rename and Delete. Projects can also be created from the File menu.
- When the WorkBench tags automatically, it now *always* matches text exactly (subject to the case and accent options you choose). The 'match from beginning' option, formerly on the Tagging Options dialog box is active only for manual tagging: the Lookup dialog box presents terms that are exact matches as well as match-from-beginning matches.
- The Tagger is no longer part of Domino Global WorkBench, and is not supported by Lotus.

Upgrading existing Domino Global WorkBench projects

The following paragraphs explain how to use Domino Global WorkBench Release 5.0.5 with projects that were created using an earlier version of Domino Global WorkBench (5.0.0 or later).

Glossaries

The design of the Domino Global WorkBench glossary has been enhanced. To upgrade an existing Release 5 glossary, refresh its design from the new template, DGWGLOSS.NTF:

1. In Notes, select the glossary database icon.
2. Choose File - Database - Refresh Design.
3. Select Local, or the server where the new template has been installed.
4. Click OK.

To upgrade a glossary from Notes Global Designer Release 4.6, you must follow the usual upgrade process.

Caution The new glossary template only works with Release 5 of Notes. If you want to continue using Notes Release 4.6 to work with glossaries, see "Release 4.6 issues" in Chapter 4.

The project database

The design of the Domino Global WorkBench project database (PROJECT.NSF in the program directory) has been enhanced. To upgrade an existing Release 5 project database, refresh its design from the new template, DGWPROJ.NTF.

1. In Notes, select the project database icon.
2. Choose File - Database - Refresh Design.
3. Select Local, or the server where the new template has been installed.
4. Click OK.

Synchronization (now called 'language synchronization')

Synchronization has been enhanced in a number of ways, notably:

- The Language Synchronizer is now a server task.
- Unilingual synchronized databases no longer have to be installed in one directory: if required they can be installed in different directories and on different Domino servers.

If you have existing synchronized databases built using an earlier version of Domino Global WorkBench, and you want to take advantage of these features:

1. Stop synchronization by disabling the Synchronizer agent.
2. After installing Domino Global WorkBench Release 5.0.5, build a corresponding template (.NTF file) for each of your existing synchronized databases. Build these templates from your original source database and glossaries.
3. Use the WorkBench to add synchronization features.
4. Replace the design of your existing databases from the templates you created in Step 2 and 3: in Notes, choose File - Database - Replace Design.
5. In order to maintain the language synchronization connections among your current documents, create an agent that has the formula:

```
FIELD $Lng_UNID := "X" + $Lng_UNID;
```
6. Run the agent on all the documents in your databases.
Caution It is important that you run this agent on *all* documents, and that you run it *after* the Synchronizer agent has been stopped and *before* the Language Synchronizer runs.
7. Install the Language Synchronizer, and set up a synchronization set using Smart Setup in the synchronization configuration database.

Upgrading Notes Global Designer Release 4.6 projects

To upgrade an existing Notes Global Designer Release 4.6 project for use with Domino Global WorkBench Release 5:

1. Create a Release 5 version of the source database. To do this just create a new **replica** on a Release 5 server -- no other change is required.

Note If you have only a tagged version of your source database, you can create an untagged version of it by using the Notes Global Designer unpopulate function (create a backup copy of the tagged database first). Domino Global WorkBench Release 5 cannot use your existing tagged database.

2. Create a copy of the relevant glossary and upgrade it. See "Upgrading an existing glossary" in Chapter 4. Having upgraded your glossary, you have two choices:
 - Use the upgraded glossary directly, or
 - Create a new, empty glossary and use it to create a tagged database (as explained in the next step). Then use the export, import and retrieve functions in the glossary to recover translations, comments and 'Prevent Translation' marks from the original, upgraded glossary.

The second option requires more work, but the glossary entries will be ordered so that translators can see all text in context when they translate, and it will contain no obsolete text.

3. Use Domino Global WorkBench to create a tagged database from the new replica of the (untagged) source database and the upgraded glossaries. A design synopsis of your existing tagged database may help with this process. See "Creating a tagged database" in Chapter 6.

For maximum re-use of existing glossary entries, when choosing tagging options:

- Do not select 'Combine Broken Terms' or 'Rich Text Sensitive'.
Note This is recommended only if it is important to make best use of existing terms, because it will prevent the WorkBench from handling pieces of rich text as units. See "Handling attributed text (rich text)" in Chapter 6.
 - Consider whether to switch on the tagging of hidden design elements. In Notes Global Designer 4.6 these were automatically tagged. In Domino Global WorkBench Release 5 tagging of hidden elements is off by default.
4. Build new language databases, **maintaining replica ID**. See "Building language databases" in Chapter 6.
 5. Finally, to maintain support for users who have only a Notes 4.6 client, deploy the new versions of the databases on a Release 5 server and set up replication between the Release 4.6 and Release 5 databases, **disabling replication of design**.

Release 5 users can now access the Release 5 version of the database. Release 4.6 users can continue to access the Release 4.6 version. Eventually all the R4.6 versions can be phased out.

Using Domino Global WorkBench with Notes Release 4.6 databases

You can use Domino Global WorkBench Release 5.0.5 to build Notes Release 4.6 language databases provided that:

- You build only unilingual databases (not multilingual ones), and
- You do not add language synchronization features.

See "Language databases" and "Synchronized databases", above.

To build Notes Release 4.6 databases you must:

- Run the WorkBench with Domino Designer Release 5 (the WorkBench is not compatible with earlier releases of Notes).

- Put the source database on a Notes Release 4.6 server. The database must be readable by Notes Release 4.6 (it must not, for example, contain Notes Release 5 design elements).
- Put the glossary or glossaries on a Notes Release 4.6 server.
- After making backup copies where necessary, replace the design of the glossaries on the 4.6 server using the '4.6 Compatible DGW Glossary' template, DKGLOSS46.NTF, shipped with Domino Global WorkBench.

Notes If you want to use a glossary that you have used for building Release 5 databases, create a copy of the glossary and use the copy for building the Release 4.6 databases. If you want to use a glossary from Notes Global Designer Release 4.6 you must first upgrade it, and you must carry out the upgrade process with the database installed on a Notes Release 4.6 server. See "Upgrading an existing glossary" in Chapter 4.

- Translate the glossary or glossaries using a Notes Release 4.6 client (you may use a Notes Release 5 client to translate plain text entries -- not rich text or graphic entries -- but it is simpler to use a Release 4.6 client for all entries).

Caution If you are using a copy of a glossary that you have used for building Release 5 databases, you *must* retranslate the rich text and graphics entries using a Release 4.6 client (the Release 5 formats are not recognized by Notes Release 4.6).

- Create the tagged database on a Notes Release 4.6 server.
- Build the language databases on a Notes Release 4.6 server.

LotusScript

When using the WorkBench with Notes Release 4.6 databases, you may encounter new LotusScript recompile errors. This is because Domino Designer uses a new release (4.0) of LotusScript which has more rigorous syntax checking routines than previous releases. You will have to correct any LotusScript errors in order to create the language databases.

To help avoid recompile problems, avoid using any of the following words as names for variables, functions, and so on. They are reserved keywords that are new to LotusScript 4.0:

ArrayAppend	FullTrim	MidC
ArrayReplace	GetThreadInfo	MsgDescription
ArrayGetIndex	%If	MsgText
CodeLock	IMESetMode	Published
CodeLockCheck	%Include	RightC
CodeUnlock	InStrC	Server
CreateLock	IsA	Sleep
DateSerial	LeftC	StrConv
DestroyLock	LenC	StrLeft
%Else	LoadMsgFile	StrLeftBack
%ElseIf	MidBP	StrRight
%End	MidBP\$	StrRightBack

Support for bi-directional languages

Domino Global WorkBench includes bi-directional functionality for the Hebrew and Arabic languages. When building a localized Domino application that includes one of these languages, the application structure will be altered to reflect the new bi-directional enhancements in Domino/Notes Release 5.

Domino Global WorkBench recognizes whether segments of translated text are bi-directional or not, and builds them out accordingly, so it is possible to include mixtures of mono and bi-directional text within individual glossary terms.

Although Domino Global WorkBench supports the localization of Domino applications into bi-directional language versions, it is not itself enabled for bi-directional display. So text may appear to be misaligned within the WorkBench and glossary. This is a local display issue, and will not affect localized bi-directional applications.

Glossary export files may display unpredictably when viewed with a text editor or operating system which has bi-directional enablement. These files are written/read using a logically-based order, while a bi-directionally enabled editor will reorder text for display based upon a visual order.

All objects in navigators, and paragraphs in rich text fields including tables, layout regions, fields and so on, will have their reading order attribute reset to right-to-left. This happens regardless of whether the localized text contains bi-directional text, or whether the term has any translation in a loaded glossary. The following are affected:

- View Columns
- View Column Headers
- Elements contained in layout regions
- Actions
- Print headers and footers for individual notes and documents
- Tables and embedded tables
- Outline headers and entries
- Embedded outlines
- View reading order

The HTML direction attribute ("DIR = RTL") will be inserted into HTML tags as appropriate. This change will only occur in:

- Paragraphs which have the pass-thru HTML text attribute set.
- Pages marked 'Treat page contents as HTML'.
- Fields named 'HTML'.
- Rich text fields which are flagged 'Treat contents as MIME/HTML' and the content-type is text/html.
- Forms marked 'Treat form contents as HTML'.

If a glossary is edited, using a bi-directionally enabled editor, prior to importation (either manually or automatically) care must be taken to ensure that this logical storage order is not disrupted. If it is, then the import will not function correctly.

The bi-directional processing included in Domino Global WorkBench will only work when building localized databases in a bi-directional language. If a Domino application is being translated from a bi-directional language to a mono-directional language (such as French or English), then some bi-directional data will still be present. Generally, this should not present a problem as Domino/Notes Release 5 will ignore such data unless it is specifically bi-directionally enabled. No data loss will occur because of this, but paragraph justifications and other alignment information may need to be manually reset using the Domino Designer client.

Bi-directional support when building Notes Release 4.6 databases

Notes/Domino Release 5 stores bi-directional data in rich text fields differently from Notes Release 4.6. Therefore, if a Notes 4.6 database is localized to produce a bi-directional version, any bi-directional text built out into a rich text item (such as form bodies) will not be visible in Notes 4.6 regardless of whether the text comes from a rich text glossary, plain text glossary, or is edited using 4.6 because Domino Global WorkBench updates terms to the Release 5 format during building.

To overcome this problem Domino Global WorkBench bi-directional processing must be disabled. You can do this by creating a new language using a language ID which is not in the form HExx or ARxx (HExx being Hebrew, and ARxx being Arabic). Domino Global WorkBench should then build all text in the standard format acceptable to Notes 4.6.

Rich text glossary terms must not be used when building bi-directional language databases for a Notes 4.6 server -- uncheck "Rich Text Sensitive" on the Lookup panel of the Tagging Options dialog box when you create the tagged database.

Setting up

The following sections explain how to get started with Domino Global WorkBench.

System requirements

The following hardware is recommended:

- A Pentium® or equivalent processor
- A minimum of 32Mb of memory
- About 32Mb of hard disk space to install the software (26Mb for a minimum install).

Operating system for running the WorkBench:

- Windows 95®, Windows 98®, Windows 2000® or Windows NT® Release 4.

Lotus software:

- To use the WorkBench:
Domino Designer® Release 5.0 or above.
- To translate Domino Global WorkBench glossaries:
Notes client, Release 5.0 or above.
- To use databases built by Domino Global WorkBench:
Server: Domino® Release 5.0 or above.
Client: Notes client, Release 5.0 or above, or a Web browser (see below).

- To install the Language Synchronizer:
Domino server, Release 5.0 or above.
- To build Notes 4.6 language databases with the WorkBench:
Server: Domino Release 4.6.
To run the WorkBench: Domino Designer Release 5.0 or above.
Glossary translation: Notes client Release 4.6 recommended.

Browsers

The browser can be any browser capable of supporting HTTP language detection, for example Netscape Navigator® 3.0 or later, or Microsoft Internet Explorer® 3.0 or later.

Installation

To install Domino Global WorkBench:

1. Run SETUP.EXE and follow the instructions that appear on the screen.
2. If you have an existing project database, PROJECT.NSF, replace its design from the newly-installed template, DGWPROJ.NTF, in your Notes data directory.
3. For Domino Global WorkBench Release 5.0.5, the glossary design has been enhanced. To upgrade the design of an existing Domino Global WorkBench glossary, select it in Notes and choose File - Database - Refresh Design.

Notes The new design uses Notes Release 5 design features and will not work with Notes Release 4.6. If you want to continue using Release 4.6 for translation work in glossaries follow the guidelines in "Release 4.6 issues" under "About glossaries" in Chapter 4. To use glossaries from Release 4.6 of Notes Global Designer with Release 5 of Domino Global WorkBench you must first upgrade them. See "Upgrading a glossary from Notes Global Designer 4.6" in Chapter 4.

4. Ensure that your Notes Release 5 program directory is named in your system path statement.

See also "Installing the Language Synchronizer" in Chapter 8.

Unicode

Unicode is a standard and universal way of encoding characters. Every character in every language is represented by its own unique number (the code point), whatever the platform, whatever the machine. This makes it possible to enter and display text in many different languages simultaneously using a single Unicode font. Usually, Unicode fonts can only display a subset of language scripts. For example:

- If you install the font MTSANS_J on the Notes client (see below), you will be able to enter translations into the Domino Global WorkBench Glossary in Japanese, in most languages that use the Latin alphabet, as well as in Greek, Cyrillic, Slavic, Turkish, Arabic and Hebrew. Your Japanese, German, Russian and Greek translations will all display correctly, and the user interface of the Glossary will continue to display correctly in English. Similarly, the font MTSANS_T gives you access to traditional Chinese characters.

- If you use suitable Unicode fonts when designing a source database, translators will be able to translate its text into all required languages without having to change the fonts to ones that can display their character set: replacing fonts throughout the design of a database is a time consuming and error prone business. However, before using Unicode in your design, you should make sure that potential users of the database have access to the necessary fonts.

The WorkBench can display text using Unicode fonts, and it will *automatically* use a font that is appropriate for the language of the text it is displaying. If you have a preference for a particular Unicode font, enter its name in NOTES.INI (in the Notes program directory):

```
DGWFont=name_of_preferred_font
```

The WorkBench will use the preferred font whenever it can. However, if it has to display text in a language not supported by that font, it will switch automatically to another Unicode font that can display the text, if one is available.

To install Unicode fonts on the Notes V5 client:

1. Run APPS\FONTS\FONTS.EXE from the Notes client or Domino Designer CD. This will unzip the following Unicode Monotype WorldType™ fonts:
 - MTSANS_W.TTF for the WGL4 character set. This font is suitable for use with most uses of the Latin alphabet (including all diacritical marks), as well as Greek, Cyrillic, Slavic, Turkish, Arabic and Hebrew languages.
 - MTSANS_J.TTF for the WGL4 character set and the Japanese language with some support for other Asian languages.
 - MTSANS_K.TTF for the WGL4 character set and the Korean language with some support for other Asian languages.
 - MTSANS_S.TTF for the WGL4 character set and Simplified Chinese with some support for other Asian languages.
 - MTSANS_T.TTF for the WGL4 character set and Traditional Chinese with some support for other Asian languages.
2. Copy the fonts you need to your WINDOWS\FONTS directory.

To enable Unicode display on the Notes V5 client:

3. Choose File - Preferences - User Preferences.
4. Under Additional Options, check 'Enable Unicode display'. This allows you to:
 - Enter Unicode characters using any installed Unicode font.
 - Display Unicode characters that have been entered using a Unicode font that is installed on your machine.
5. Click Default Fonts. Select Unicode fonts for 'Default Serif', 'Default Sans Serif' and/or 'Default Monospace'. This allows you to:
 - Enter Unicode characters using the Notes 'Default' fonts.
 - Display Unicode characters that have been entered using the Notes 'Default' fonts.

To enable Unicode on a browser:

- Install a Unicode font.
- Select the Unicode font for use in the browser (in Internet Explorer 5 choose Tools - Internet Options and click Fonts).

- Install language support files -- usually you will be prompted to do this when you first access a page that needs it. Follow the on-screen instructions.

For more information about Unicode, see <http://www.unicode.org>.

Documentation

The Domino Global WorkBench documentation is available in three formats:

1. As a Notes database (DGWHELP.NSF), for access when working in Notes.
2. As a Windows Help file (DGWHELP.HLP), for access when using the WorkBench. This file provides context-sensitive Help for the WorkBench.
3. This manual, *Guidelines for Developers and Translators*, released in Adobe® Acrobat® (.PDF) format. It is available from the Lotus Web site at www.lotus.com/international.

For detailed information about the various screens and dialog boxes, refer to the "Reference" section of the Notes database or Windows Help file.

More information, tips, advice . . .

For additional information about using Domino Global WorkBench, plus valuable tips, advice and discussion see the Template Globalization Best Practices database (V5TemplateLocalization.nsf). To access this go to <http://www.lotus.com/international>.

For Notes, Domino and Domino Designer documentation, including translated versions, go to <http://www.notes.net/notesua.nsf>.

Chapter 2

Recommended processes

This chapter outlines some recommended ways of using Domino Global WorkBench. Details of the various steps are given in the chapters that follow.

Localizing Domino databases

The process described here is suitable if you are starting from scratch, with only a database that needs to be localized. If you want to build synchronized databases you will also need to look at the next section, "Creating synchronized databases". And if you want to include translation workflow see "Adding translation workflow".

Step 1: Prepare the source database

The source database is the one you want to localize (translate). There are a number of things you can do that will simplify the localization process, and these are described in Chapter 3. The more important ones are mentioned below.

If the database is still being designed:

- Pay attention to general design considerations that will make the translator's task simpler.
- Use aliases wherever you can, and use a system of standard prefixes for them, so that the WorkBench will recognize them.

If the database has already been designed:

- Make a backup.
- Create a design synopsis and mark on it all design elements that need aliases.
- With the help of the design synopsis, create aliases. Use a system of standard prefixes for aliases, so that the WorkBench will recognize them.

Note The WorkBench tagging operations will operate more accurately and with fewer errors if all text that can be aliased is aliased.

See Chapter 3, "Preparing source databases".

Step 2: Create a glossary

The glossary will hold the terminology that is used in the source database. To prepare a new glossary:

1. Create a new glossary database in Notes.
2. As reference language select the language of the source database.
3. Do not select any 'languages to enable' at this stage: your reference language (the language of the source database) will be the only active language.

At this stage the glossary is simply an empty database. The WorkBench automatically fills it with entries (which will later be translated) when it creates the tagged database. New glossary entries can also be made direct from Notes.

Notes If you wish you can delay this process until you load the glossary in Step 3 (there is a Create button on the Select Glossary dialog box). If you have an existing glossary you can use that rather than create a new one (if it is from Notes Global Designer Release 4.6 you will need to upgrade it).

See Chapter 4, "Working with glossaries".

Step 3: Start the WorkBench

Start the WorkBench from the Domino Global WorkBench folder, then:

1. Create a project. In Domino Global WorkBench, a 'project' is a set of one or more source databases, together with their associated glossaries, tagged databases and language (output) databases.
2. Load the source database (the one to be localized).
3. Load the glossary. Once the glossary has been loaded you will see the contents of the source database in the Design element and Name panes to the right of the display.

Notes You can load more than one source database if you wish. You can also load more than one glossary.

See Chapter 6, "Using the WorkBench".

Step 4: Create the tagged database

When working with a new source database and a new glossary, the following process is recommended for creating the tagged database:

1. In the WorkBench, select (tick) all the source database's design elements for tagging (in the Name pane, with the source database selected).
2. Choose Create Tagged Database from the Actions menu.
 - Supply a server and file name for the tagged database.
3. Choose the following tagging options:
 - On the Basics panel: Term Type (select 'No term type'); Create New Terms in This Glossary By Default (select the glossary you created in Step 2).
 - On the Lookup panel: Case Sensitive and Accent Sensitive (both checked); Combine Broken Terms and Rich Text Sensitive (both checked). Database name, Package, Class and Property (all unchecked); Type, Type title, Sub type, and Sub type title (all checked).
 - On the Prompting panel: Never prompt (selected); Create new terms (checked); As needed (selected); Always prompt when tagging formulas (unchecked);
 - On the Tagging panel: Tag the following and Tag (both unchanged). If you created aliases in the source database, and if you used standard prefixes (highly recommended, see Step 1), add a wildcard entry in the Exclusion list for each prefix, for example va_* if you are using 'va_' as a prefix to identify view aliases.
4. Click OK.

This automatically creates the tagged database and creates entries in your glossary, based on the text in your database.

Notes Other tagging options will be appropriate for other circumstances. See "Creating a tagged database" in Chapter 6.

Step 5: Review the glossary

Prepare the glossary for translation and building:

1. Add comments to help the translators. Good commenting can have a very significant effect on the speed and quality of translation.
2. Look for any terms that should not be translated. Typically these will be:
 - References to aliases in formulas, LotusScript or JavaScript. They must not be translated if functionality is to be preserved. Typically these are references to forms, views, agents, and the like. For example, in a formula such as `@command([ChangeView];"mainview")` the text `mainview` is usually an alias. If your source database uses aliases, the aliases have standard prefixes (see Step 1), and you included the prefixes in the WorkBench's exclusion list when creating the tagged database (see Step 4), you should have few problems here.
 - Text strings that never display, and therefore do not need to be translated. For example, the names of hidden views. If you created a design synopsis (see Step 1), you will find it useful here.
3. Mark the terms that should not be translated (DNTs) as 'Prevent Translation'.

Note It's a good idea to keep entries in the glossary for these 'do not translate' terms because you will then not have to think about them again, for example when processing a revised version of the database. You could leave them untagged, with no glossary entry, but you would then encounter them again, as 'new' terms, when you created a tagged database from a new version of the database.

See "Text that must not (or need not) be translated" in Chapter 5.

Step 6: Create 'pseudo' translations

Domino Global WorkBench glossaries support a 'pseudo' language which allows you to test the process before embarking on translations. Pseudo-language glossary entries are created automatically from the reference language entries. They are composed of meaningless text, designed to simulate translation.

1. In the glossary, select 'Pseudo' as an active language. The reference language (the language of the source database) is already active.
2. The 'pseudo' language option allows you to test the process of creating the tagged and language database(s) before you start any translation work.

You create pseudo translations in the glossary by clicking 'Pseudo Translate / All documents or Selected documents' on the main glossary display.

Notes You can have other languages active at the same time as 'Pseudo', but at this stage only the reference language and 'Pseudo' are needed, and processing is quicker if only two languages are active.

The pseudo-language translation also lets you see the effect that text expansion has on the layout of your database (text can expand by 30% on translation).

See "Creating pseudo-translations" in Chapter 4.

Step 7: Build a pseudo-language database

This creates a language database that has its text in the single language 'Pseudo'. The process is the same as for building any language database:

1. Define the database.
 - Select the single language 'Pseudo' for your database.
2. Select the elements for building.
 - Select all the elements in the database.
3. Create the database.

See "Building language databases" in Chapter 6.

Step 8: Test the built database

The pseudo-language database created in Step 7, although its text will look strange, should function exactly as the original source database, and:

- If you chose to expand the text when creating the pseudo-language entries in the glossary you will see the effect of that expansion on the layout and general appearance of the built database.
- All visible text should appear pseudo-translated. Any text you see in the original language, unchanged, either has not been tagged in the tagged database, has been marked 'Prevent Translation' in the glossary, or has been deselected from the build process.

As a result of testing the built pseudo-database, you may need to make changes:

- To your source database. For example:
 - To add an alias where one was missing, or make a formula refer to an alias rather than a design element name.
 - To adjust the layout to take account of text expansion. This applies especially to tables and navigators. If expansion is not allowed for in the source database, the layout of every language database built from it is likely to need adjustment.
- To the Prevent Translation flags in the glossary. Typically, you might want to add a Prevent Translation flag because:
 - Some text that must not be translated has appeared in the pseudo language in the built database, or
 - A reference to an alias has mistakenly been translated. This will affect the proper functioning of the built database.
- To the tagging in the tagged database. In this case it is often simplest to create the tagged database from scratch again, but with different prompting options so that you can choose whether to tag (or not to tag) the problem text.

See "Tagging" in Chapter 6.

Step 9: Translate

Once you can build a correctly functioning pseudo-language database, you can start translation:

1. If the source database has changed since you started work:
 - If the changes are less than minor, create a new glossary (Step 2) and a new tagged database (Step 4). Use the export and import functions to move comments and DNT information from the original glossary into the new one. This ensures, firstly, that translators can access text in its natural order (for example all pieces of text on a form together and in order) and, secondly, that there is no obsolete text in the glossary.
 - If the changes are minor, you can simply update the tagged database (items 1 and 2 of Step 11), although you will then find that a few pieces of text in the glossary are out of their natural sequence.
2. In the glossary, make active the languages into which you will be translating.
3. Translate the glossary entries.

You can now build language databases.

See Chapter 5, "Translating", and "Active languages" in Chapter 4.

Step 10: Build language database(s)

This creates the language database(s) that people will use. The steps are the same as for creating the pseudo-language database in Step 7:

1. Define the database.
 - Select one or more languages.
2. Select the elements for building.
 - Select all the elements in the database.
3. Create the database.
4. Repeat the process if you want to create other language databases.
5. (Optional) Add language synchronization features to your built databases. See Creating synchronized databases for a description of the process.
6. (Optional) Add translation workflow to your built databases. See Adding translation workflow features for a description of the process.

See "Building language databases" in Chapter 6.

Step 11: Update

If design changes are subsequently made to the source database, the WorkBench can update your tagged and language databases to reflect the design changes:

1. Select the source database and choose View - Refresh to see the changes in the WorkBench.
2. Click 'Update Tagged Database' to bring the tagged database into line with the new source database. New glossary entries may be created during this process.
3. Translate any new glossary entries.

4. Select each language database in turn and click 'Update Language Database' to rebuild them from the new tagged database and glossary.

Notes When you run the update, WorkBench rebuilds the design elements that have changed. For forms, it rebuilds only the fields that have changed. However, if the original design element, for example form, has been *re-created* (for example by copying and pasting) rather than just edited, it will have a different unique ID and WorkBench will treat it as a completely new design element.

See "Updating" in Chapter 6.

Creating synchronized databases

The process described here is suitable if you want to add language synchronization features to a multilingual database or to a set of unilingual databases. The databases must have been built by Domino Global WorkBench, and they must have been built from the same source database. For more information, see "Implementing language synchronization" in Chapter 8.

For a basic process for localizing the design of a database or template, see "Localizing Domino databases", at the beginning of the chapter. And if you want to include translation workflow see "Adding translation workflow", below.

Step 1: Select the language databases

'Language synchronization' allows you to automate the process of translating documents in built language databases. The language databases might, for example, constitute an international Web site.

In a synchronized database documents are flagged as local, global or translatable. 'Local' documents are relevant only to the language they were created in. 'Global' documents are relevant to all languages, but they don't need translation. 'Translatable' documents are relevant to all languages, and they do need translation.

A scheduled server task, the Language Synchronizer, automatically copies new global or translatable documents to other languages in the 'synchronization set' so that they can be seen by other users and translated where necessary.




To add language synchronization features:

1. Start the WorkBench and open the project that contains the language databases you want to synchronize.
2. Choose View - Switch to Synchronization to go to the language synchronization display.
3. Display the language database icons by selecting the source database from which they have been built.
4. Put a tick mark against each language database that you want to add language synchronization features to.

If your databases are unilingual you will need to select more than one of them to form a 'synchronization set'. If your database is multilingual, select it alone.

Step 2: Select the synchronization type

To allow documents to be synchronized, the WorkBench adds a hidden subform to the relevant forms in the language database(s). You must specify which forms are to be synchronized and their default synchronization type. On the language synchronization display:

1. Select one of the language databases (it doesn't matter which one). Its forms are displayed in the pane on the right.
2. Click in the Title column next to each form to select:
 - Local, , if documents created from the form should not, at least initially, be viewed in other languages. If you allow users to change the synchronization type of documents (see Step 3), local is a good setting to choose here. Users can then keep their own documents local while they write and edit them.
 - Global, , if documents created from the form should be passed to other languages, but do *not* need translation.
 - Translatable, , if documents created from the form should be passed to other languages, and *do* need translation.
 - No selection if you do not want the synchronizer ever to process documents created from the form.

Step 3: Set user options

You can specify whether or not users are allowed to change the synchronization type (local, global or translatable) of documents they create from the various forms. For each form, click in the 'Notes' and/or 'Web' columns to add or remove ticks:

- A tick in the 'Notes' column means that users working from a Notes client will be able to select a synchronization type when they create a document using this form.
- A tick in the 'Web' column means that users working from a Web browser will be able to select a synchronization type when they create a document using this form.

Step 4: Add the synchronization features

To add the language synchronization features to the database(s):

1. Click Synchronize Languages or choose Actions - Synchronize Language Databases. A dialog box is displayed.

If you are synchronizing unilingual databases you can choose to include a Language SwitchBar on translatable documents. To do this select Yes for 'Add Language SwitchBar'. A SwitchBar will then be added to all translatable documents created in the synchronized databases: users click on its SwitchBar to view a document in another language.
2. Click OK. The language synchronization features are added to (or updated in) all the ticked language databases, subject to the selections in the 'Type', 'Notes' and 'Web' columns.

Step 5: Deploy the database(s)

Once you have added the language synchronization features to the language databases, you can deploy the databases to servers so that users can access them. To do this:

1. Check that the Language Synchronizer server task has been installed on a suitable Domino server. Language Synchronizer installation is a job for the server administrator. As part of the installation, the Synchronizer configuration database is installed in the server's Domino data directory.

2. In the WorkBench, open the project that contains your synchronized databases, as created in the previous Steps.
3. If you are not there already, go to the language synchronization display by choosing View - Switch to Synchronization.
4. Put a tick mark against each synchronized language database that you want to deploy.
5. Click Configure Synchronization Set and complete the Configure Synchronization Set dialog box. This puts the necessary information about the synchronization set, and the required synchronization schedule, into the Synchronizer configuration database.

Note Configure Synchronization Set assumes that the synchronized databases will reside in the data directory of the Domino server where the Language Synchronizer server tasks runs. If this is not the case, you will need to make changes in the Synchronizer configuration database.

See "Using the Synchronizer configuration database" in Chapter 8.

6. Deploy the databases on the required server(s).

Adding translation workflow

The process described here is suitable if you have a set of synchronized databases (or a synchronized multilingual database), typically a Web site, and you want to include translation workflow features (see Chapter 8). For a basic process for localizing the design of a database or template, see "Localizing Domino databases", at the beginning of the chapter. For a process for creating synchronized databases, see "Creating synchronized databases", above.

Step 1: Preliminaries

Before applying translation workflow to a set of Domino databases:

1. Prepare your source database, making sure that it contains the views you need.
2. Use the WorkBench to build the working (language) database(s) from your source database, and then implement language synchronization. This can be a single multilingual database or a set of unilingual databases.

The Domino Global WorkBench translation workflow application works *only* with synchronized databases created by the WorkBench.

Step 2: Create the workflow control database

The translation workflow database will hold:

- Control documents for the workflow process
- A tracking document for each document being translated

Translators use the translation workflow database to see and change the status of the documents in translation and to access those documents (via doc links in the tracking documents). To create the translation workflow database:

1. From Notes Release 5, choose File - Database - New.

2. Supply the following information:
 - The server for the database. This must be the Domino Release 5 server where the Language Synchronizer and Synchronizer configuration database have been installed (see Step 5 of "Creating synchronized databases", above).
 - A title and filename for the database.
 - The workflow template, DGWWFLOW.NTF.
 3. Click OK.
- At this stage the database will be empty.

Step 3: Assign roles and access rights

The people who will be working on the translations need the appropriate access rights.

1. In each of the synchronized databases where the workflow will operate give yourself, and each of the other people who will play a part in the translation process, editor access. Give yourself, and anyone else who will be managing the translation workflow database, 'Delete documents' rights.
2. Select the new translation workflow database in Notes and choose File - Database - Access Control. Give yourself, and anyone else who will be managing the translation workflow database, at least editor and preferably manager access. Make sure that 'Delete documents' access is checked. Add the names of the other people who will play a part in the translation process, giving them editor access.

Step 4: Create the workflow information document

The workflow information document provides basic information for the workflow process.

1. In Notes, choose Create - Workflow Information (or click Yes if Notes prompts you to create a workflow information document).
2. Complete the workflow information document. You must supply the following information:
 - The name of the 'synchronization set' for which you are enabling translation workflow. This is the group of unilingual synchronized databases, or single multilingual one, to which you are now adding translation workflow. Your synchronization set should already have been defined in the Synchronizer configuration database, and you should therefore be able to select it from the drop-down list.
 - The name of the 'webmaster'. This is the manager of the translation workflow database, and usually also the manager of the synchronized databases.
 - The name of a field (the 'Hint fieldname') whose contents can be used to identify documents in views in the translation workflow database. This is a field that will suitably identify your documents when they are listed in the translation workflow database. It will typically be Subject or Title.
3. Click 'Retrieve Synchronizer Data' to enter automatically the languages and databases in the synchronization set.

When you save the workflow information document, an agent automatically creates language information documents, one for each language used in your database(s).

See "Setting up translation workflow" in Chapter 8 for more information.

Step 5: Edit the language information documents

When you save the workflow information document, an agent automatically creates language information documents, one for each language used in your database(s). These documents specify who will be the default translators and reviewers for translations into the respective language.

1. Open each of the language information documents. They are held as responses to the workflow information document.
2. Select Yes or No for 'Translation manager review'. If this is set to Yes the translation managers will receive a mail whenever a new, translatable document is created for their language pair. He or she can then review the job assignments.
3. Click Workflow Assignments and specify the names of people who will, by default, fill the roles of Translation Manager, Translator, Reviewer, Approver and so on for documents authored in other languages.
4. Save and close.

See "Setting up translation workflow" in Chapter 8 for more information.

Step 6: Schedule the workflow agent

To schedule the workflow agent:

1. In the translation workflow database, choose View - Agents. Domino Designer opens.
2. Open the Workflow agent.
3. Under "When should this agent run?" select a running frequency -- usually 'On Schedule More Than Once a Day' or 'On Schedule Daily'.
4. Click Schedule and set a suitable schedule. At a minimum you should run the workflow agent each time the Language Synchronizer has completed its operation.
5. Save the agent document.

See "Setting up translation workflow" in Chapter 8 for more information.

Chapter 3

Preparing source databases for localization

This chapter explains how to design a database that will be as simple as possible to localize. If you pay attention to these guidelines, the localization task will be smaller, quicker and will cost less.

Note For synchronized databases, some additional preparation steps are usually needed. These are described in Chapter 8.

International considerations

When designing database applications for international environments, keep the following points in mind.

Acronyms

Acronyms can be very convenient when creating an application and are often used to save space on a form or view. In general they will need to be translated, like any other piece of text. Bear in mind that:

- There may be no corresponding acronym in the target language so the text will have to be represented in full.
- Sometimes, acronyms have negative connotations in other languages.
- Translators will need to know what the acronym stands for. Add an explanatory comment in the application itself or (better) in the glossary.

Here are a few examples of familiar acronyms. To an English speaker the translated versions are not recognizable:

OAS: Organization of American States

OEA: Organización de Estados Americanos

UNO: United Nations Organization

ONU: Organisation des Nations Unies

US: United States

EU: Estados Unidos or États Unis

Colors

Avoid bright colors, especially red, purple, yellow and other 'hot' colors. These tend to mean different things in different cultures (for example, in the US red often signifies 'stop'). Black boxes around people's names are not acceptable in some cultures. Remember that color differences will be lost when printing to a black-and-white printer (red will print as black, for instance).

Culture

Be careful not to offend your customers. References that are not considered offensive in your country may be so in other countries. Your users are likely to be scattered across the world, belonging to many religions and races, and holding many different beliefs.

Examples that make good sense in one country may not be understood at all in another. An example involving a knowledge of baseball or cricket would not make sense to many people.

Expansion

You are likely to encounter expansion problems if your working language (the language in which your database application is developed) is one of the 'shorter' languages, such as English. Translated text can be up to 30% longer than English (more, if the text is just a word or two), so if you are working in English try to leave space for expansion when you design forms, views and so on.

Text expansion often causes problems in tables. Translated tables often have to be reformatted, and this can be very time-consuming. If you need to use tables, design them with expansion in mind.

The length of currency symbols varies from country to country. For example, US and UK currency symbols are single characters (\$ and £), but in Italy the currency symbol is sometimes represented as 'Lit. ' (five characters). The sizes of numbers will vary from country to country. For example, a house might cost £100,000 (eight characters) in the UK: the equivalent in Italy could be Lit. 200,000,000 (16 characters). In a case like this you might need to leave space in your design for a heading such as 'All numbers in 000s'.

Expansion - Action buttons

Action buttons are sized dynamically by Notes. If you display several of them, using close to the full screen width, you may find that, after translation, the rightmost one or two will disappear from view. If you are designing in English, consider the effect of a 30% expansion of button text on translation. To save space you can omit graphics from actions buttons, leaving only a text label.

In Notes Release 5 you can use the following features to overcome these problems:

- Cascade the buttons. This groups them under one root button, in the same way as action names can be grouped under one Actions menu entry. Simply use a common root name for the buttons you want to cascade, for example Tools\Create Group and Tools\Categorize.
- Include slider controls on action bars. This is done via an infobox. It allows buttons that do not fit on the screen to extend beyond the end of the standard action bar.

Fonts

When designing a database, check that the fonts you use are available to the people who will be using your database. When Notes encounters an unknown font it uses a suitable alternative, but this may have an adverse effect on the layout and/or readability of your application. Let the translators know what fonts and font sizes you plan to use. If you intend to use an unusual font make sure the translators have access to it.

If you think that your application might be localized into Asian languages, you should use fonts which are acceptable for double-byte environments, otherwise Asian translators will have to change all fonts manually in the database's design, which is a time-consuming task, especially when dealing with views and columns.

Unicode

If your database is to be localized into languages that require very different character sets (for example you might plan to localize into German, French, Spanish, Japanese and Chinese), consider using Unicode fonts. See "Unicode" in Chapter 1.

If you cannot use Unicode and your database is being developed for use with Notes Release 5, use a virtual font such as Default Sans Serif, Default Serif, Default Monospace, or Default MultiLingual. Each of these is associated with a real font in the User Preferences dialog and each language version of Notes is installed with suitable defaults.

If a character, for example an accented character, is not available in a font, an 'unknown character' symbol (a box or vertical bar) will display. For example, Japanese characters will often display as unknown characters in European environments. This is not a data corruption problem: Notes keeps your data secure and will display it properly wherever it can.

Formats

The following are typically different in different countries. On a small scale: date formats, number formats, names and postal codes. On a larger scale: formats of addresses, formats of letters.

In general Notes is aware of local formats, and so Notes clients in two different countries may display dates and numbers differently. One parameter that is set by Notes is the starting day of the week, so if you're writing formulae to perform calculations with weeks you should bear in mind that the start days will change according to the settings in the server or client machine.

Formats - Addresses

Address formats also vary widely from country to country. Some countries have states, others have provinces; some countries use five or more letters and numbers in postal or zip codes, other use numbers only; and in some countries the postal or zip code appears before the state or province, in others after. As there is no standard format, one must try to identify the minimum requirements to obtain a complete, widely accepted address. For instance:

- Provide generous space for the address
- Do not offer choices such as Rd, St, Ave or Sq. Let the user define this.
- Label the 'ZIP code' field as 'ZIP/Postal code'
 - Make it optional
 - Do not check it for validation
 - Do not provide options like state abbreviations
- Label the 'State' field as 'State/Province'
 - Make it optional
 - Do not check it for validation
 - Do not provide options like state names to choose from
- Do not try to force a set format. A good choice would be to align the options vertically, one field per row.
- Do not construct full addresses in one single view column. Just display each field in a different column.

Formats - Currencies

When using currency fields, consider using numeric rather than currency fields, adding the currency symbol as static text. The translators can then simply localize the static text (which will appear in the WorkBench glossary). If currency fields are used, the field properties of each currency field will have to be changed in each built language database. In views, the currency symbol should be included with the relevant column title.

Also, remember that Windows 95 and Windows NT 4 support the Euro currency symbol, €, only if the appropriate update has been installed. You may therefore prefer to include the symbol as a graphic or to type the full word 'Euro(s)' as text next to the relevant number fields.

Formats - Personal names

The format of personal names varies around the world. In the US a name is typically formed by a person's first name, middle initial and last name. In many other places, including the Middle East, Europe and South America, middle initials are not used. In some countries people may have two or more surnames. One way of avoiding name problems is to create a large free-form field for the whole name and label it 'Full name'. Or, if you want to be more precise, use two fields: one for the first name and middle initial if necessary, and the second for the surname or names.

Graphics

If you use graphics in your design, try to use international ones. A symbol that is very familiar in one country (for example a symbol for the rail network) may be meaningless in other countries.

Where possible, use graphics without text: they will usually then require no localization. This applies especially to animated GIF which are very time consuming to translate.

If graphics need to contain text, do what you can to help translators:

- If you are developing a Release 5 database, store the graphics as image resources. Once stored in this way they can be inserted into your design many times. This feature can be very useful for buttons, and will save file management and translation overhead.
- Make sure the source files are available in a format where the text can easily be translated. It is usually easiest for the translators if the text is held in a separate 'layer'.
- Keep any subsidiary files that may be useful, and make them available to the translators. For example, if you set up a sample database in order to make a screen shot of a particular form or view, it will often be useful for the translators to have a copy of the database.
- Allow space for expansion (see above).

Good programming practice

If you follow the programming practices described in the following sections, the localization of the database application with Domino Global WorkBench will be much simpler.

Using aliases

Wherever you can, create and use aliases. This ensures that displayed text can be translated without disturbing the functioning of the database. The WorkBench recognizes aliases and does not tag them: it tags only the displayable text.

For best results you should alias *all* design element names using the method outlined in Item 1, below. And you should alias text within design elements if that text can be referenced programmatically, for example keyword field values.

When referencing design elements in formulas or script, use the aliases whenever possible. Where you cannot use an alias, make a note so that localization groups are aware that a name is being used.

There are three ways of handling aliases:

1. The best approach is to adopt a system of unique prefixes. Then it will immediately be clear when a formula (or LotusScript) is referencing an alias. For example you will see something like this:

```
SELECT (Form="fa_New" | Form="fa_Current")
```

Here, the prefix 'fa_' is used to indicate a form name alias. For maximum accuracy of tagging and building use the prefixes recommended by Lotus and listed in the table below.

When translators see pieces of text with the recognized prefixes they will know that they are aliases and should not therefore be translated, while other references are not aliases and should be translated. Furthermore, the person creating the tagged database can make the WorkBench automatically exclude these from tagging by putting a wildcard entry, for example fa_*, into its exclusion list (on the Tagging panel of the Tagging Options dialog). Even if such aliases are tagged by mistake and find their way into glossaries they can easily be located and marked 'Prevent Translation'.

This approach requires some design work, and if left until the design is complete, is more prone to error. However, it does make aliases clear in formulas, and it allows error-free creation of the tagged database to be largely automatic.

Caution If your prefixes contain underscore characters you must exclude the underscores in the following two @Commands:

```
@Command([OpenNavigator];"navigator") and @Command([ViewChange];"view").
```

For example, if a navigator has the alias na_Main, you should enter naMain in the @Command. Then add the text "naMain" to the Exclude list on the Tagging panel of the Tagging Options dialog in order to prevent the WorkBench from tagging it.

2. A simpler approach is to make aliases exactly the same as the original text. Formulas will automatically use the alias, and the original text can safely be tagged for translation. However, there are one or two drawbacks:
 - There will be no visual indication in the formulas themselves that aliases are being referenced. For example, in the following formula the text 'New' and 'Current' could be form names or they could be aliases of form names (the only way to be sure whether or not an alias is being used is to check the form itself):

```
SELECT (Form="New" | Form="Current")
```

If everything has been properly aliased, it will be safe for translators to treat these as aliases, and either not tag them or tag them but mark their glossary entries as 'Prevent Translation'. The latter approach will only work effectively if appropriate context options are checked when the tagged databases is created. Otherwise, there is a danger that several occurrences of the term in the database will be tagged to the same glossary entry. In some places the term will be the alias, and therefore should not be translated, whereas in others it will be normal displayable text which should be translated.

Note There are cases, for example in a simple action, where a reference is *always* to a name, never its alias. See "Simple actions", below, and "Tagging" in Chapter 6.

- Each alias will be the same length as the text it is aliasing, and this may be inconveniently long, for example when working in formulas.
3. Finally, you can use numbers as aliases. Normally, where aliases are text strings rather than numbers, and do not have identifying prefixes as suggested at 1 above, the WorkBench never tags them, but it *will* tag references to them in formulas (it cannot distinguish them from other displayable text). The person doing the tagging then has to take some action: either to leave the tag in the formula text, but mark the term 'Prevent Translation' in the glossary; or to remove the tag.

However, the WorkBench *always* ignores numbers, so if an alias is a number, the WorkBench will not tag the alias *and* it will not tag references to it in formulas. The translator's task is simpler: there are no tagged references in formulas to consider.

Using numbers as aliases may not always be convenient, but it does help translators. Consider using them where possible, for example for keywords.

See "Creating aliases" at the end of this chapter.

Recommended alias prefixes

The following table lists the alias prefixes recommended by Lotus. By default, all of them are included in the exclusion list on the Tagging Options dialog box. The exclusion list entries have the form aa_*, where the asterisk (*) is a wild card.

Design element	Alias prefix
Agent	aa_
Field name*	fl_
Folder	fd_
Form	fa_
Frameset	fs_
Frame	fr_
Image resource	ia_
Java resource	ja_
Keyword field value	ka_
Navigator	na_
Page	pa_
Outline	oa_
Outline entry	oe_
Subforms*	sa_
View	va_

* Subform names are never translated because they are never seen by users. However, because they appear in computed subform selection formulas, it is sound practice to alias them so that translators can easily recognize them. Similar considerations apply to field names.

Field names

Field names, which cannot be aliased, are never displayed on the screen and so they do not need to be translated. However, sometimes they will *look* like displayable text, for example when they appear in formulas or LotusScript. You should therefore use a standard prefix so that translators can recognize them. For example:

fl_Status

By default the WorkBench has the entry fl_* in its exclusion list (on the Tagging panel of the Tagging Options dialog). The asterisk is a wildcard.

Avoiding concatenated sentences

Avoid constructing sentences using formulas such as:

```
"Today is "+@weekday(@today)+" "+@month(@today)+" "+@day(@today)+"  
"+@year(@today)
```

The length of this will vary according to the local language and setup, and in many languages the order of the elements will be different (so the translators would have to change the formula, not just translate the text). In this example you could instead use a simpler format:

Date:

where TodaysDate is a time field calculated using the formula @today. The same considerations apply to LotusScript or Java code.

And avoid doing the same sort of thing with fields, for example:

Created by on for

Translating a sentence like this may involve changing the order of the fields, and that means additional work and may introduce errors. A simpler format would be:

Created by:	<input type="text" value="Author"/>
Creation date:	<input type="text" value="Date"/>
Customer:	<input type="text" value="Customer"/>

Keywords

Keep them simple

If you use keyword fields (Dialog list, Checkbox, Radio button, Listbox and Combobox field types), try to keep the keywords simple. They are difficult to translate because the field label text is tagged separately from the keyword values, so they end up as separate glossary entries. The translator sees several, apparently unconnected, pieces of text (although s/he will have some context information to help). In many languages there may need to be grammatical agreement between them.

For example, the field:

Schedule status:

will be harder to translate than:

Schedule updated? Yes/No

Using formulas for keywords

Avoid entering keywords using a formula. Instead, use the static items option 'Enter choices (one per line)' in the keyword field properties box. The WorkBench will correctly tag keywords, provided they are static and aliased.

If you do use formulas in the keyword field, you should always include aliases, for example:

```
@if(  
  fl_house = "ka_apart"; "Apartment | ka_apartment";  
  fl_house = "ka_town"; "Town house | ka_townhouse";  
  fl_house = "ka_country"; "Country house | ka_countryhouse";  
  Estate | ka_estate);
```

If you reference the keywords in other formulas, use the aliases, not the keywords themselves. As with all aliases, you are recommended to give them a prefix so that they can be automatically excluded from the tagging process.

Note If you use the keywords to *display* text in categorized views (this does not apply to hidden views) you will have to write a formula in the view column to convert the alias into text because columns display aliases only.

Displaying in views

If you want a view to display a keyword field do not use the Field option, because that will display aliases, not the keywords themselves, and aliases are not tagged or translated. Instead use the Formula option:



Here, a keyword field ActiveValue has the following keywords and aliases:

Completed | ka_Completed

Active | ka_Active

And the prefix 'ka_' has been used to denote a keyword alias.

Numbers as aliases

Consider using numbers as aliases for keywords. See item 3 in "Using aliases", above.

Formulas and script

The WorkBench recognizes and tags text in formulas, LotusScript, JavaScript and Java. In general it tags all text it finds between standard string delimiters such as double quotes ("). To help translators:

- Use aliases when referencing other design elements. And use a system of prefixes so that translators, and the WorkBench, can recognize the aliases. See "Using aliases", above.
- In more complex formulas, exclude translatable text from the formula itself. Instead, use temporary variables to hold the text and reference the variables in the formula. Use a system of prefixes to identify variables, and use REM comments to indicate to translators the context and meaning of each text string. See "Variables", below.

Hide when formulas

'Hide when' formulas can be associated with text in a database. When they evaluate to TRUE, the text does not display. These are held in the same way as text attributes (they are accessed through the text properties box), and they are preserved by the WorkBench, provided 'Combine Broken Terms and 'Rich Text Sensitive' are selected on the Lookup panel of the Tagging Options dialog. You should always use aliases in conditional hide/show formulas.

Where you use them, make a note and inform the localization team.

See "Handling attributed text (rich text)" in Chapter 6.

Variables

Variable names do not display and therefore do not need to be tagged. However, sometimes they will *look* like displayable text, so it is advisable to use a standard prefix to distinguish them. For example:

```
fl_Action := "tmpAccept"
```

Here the prefix 'tmp' is used to denote a variable. The person creating the tagged database can make sure the WorkBench does not tag these by putting a wildcard entry, for example tmp*, into its exclusion list (on the Tagging panel of the Tagging Options dialog).

To help translators, you should use temporary variables for storing text in more complex formulas.

The use of temporary variables is especially helpful in @prompt and @dialog titles and body text. Ideally, use a convention for naming the variables, for example:

```
hd_Entry := "Data entry" for box title text
```

```
tx_First := "First name:" for body text
```

```
ky_Yes := "Yes" for a translatable keyword
```

```
db_Info := "Employee information" for a database name for a lookup
```

LotusScript

The WorkBench tags delimited text unless it is part of the LotusScript language. Standard delimiters are double quotes ("), vertical bars (|) and braces ({ }). To allow Domino Global WorkBench to handle LotusScript efficiently you should create aliases for all design elements and use standard prefixes so that the WorkBench can recognize them. See item 1 in "Using aliases", above.

To simplify localization, avoid using script in external LSS files. These will have to be localized separately.

Usually it is best to leave text in its logical place within the script. However, in some applications you may want to isolate it. There are a number of ways of doing this, for example:

- In a script library by creating a function and using Case statements to store the different pieces of text.
- By defining Public constants in the Options object of the relevant script libraries.
Note When building language databases, the WorkBench recompiles script libraries only if they are selected for building.
- In hidden forms by using computed text fields to store your text. To do this, create a hidden form and add computed text fields to hold your pieces of text. Enter the text in the Value event for each field. To help translators, use field names that give some idea of what text they hold and where it belongs. To access the text, your script creates a document in memory from the hidden form (no need to save the document) and then uses the method *Notesdocument.getItemvalue("fieldname")*, where 'Notesdocument' is the name you gave to the document.

JavaScript

JavaScript is processed by the WorkBench in the following situations:

- When it appears between <script> and </script> tags in any HTML that is processed by the WorkBench (see "HTML", below).
- Wherever native JavaScript is supported in Domino Designer. For example in views, in events or in actions placed on the action bar.

In these circumstances text strings are tagged for translation. A 'string' is any text that appears between double or single quotes.

JavaScript tagging works best when the text that constitutes it is kept as plain as possible. The presence of non-text Notes rich-text objects (for example fields, tables and buttons) or embedded subforms within a JavaScript stream make it difficult for the WorkBench to properly extract the translatable text. This is because when these objects are embedded, the JavaScript becomes fragmented into many smaller pieces which are difficult for the WorkBench to analyze.

Caution The comments in the previous paragraph are especially true of JavaScript in HTML. If the WorkBench cannot correctly process the JavaScript, it is likely that it will also fail to process the surrounding HTML correctly.

It will help the translators if you put all the text in one place. To reference text in JavaScript use a function to fetch it from a script library or a hidden form. See "LotusScript", above.

Handling text that never displays

Once a database application has been developed, it is the job of an experienced engineer and a translation specialist to prepare it for translation (see "Who does what?" in Chapter 1). Part of this work is to ensure that text that must not be translated is marked as such in the glossaries so that translators do not waste time translating it (see "Terms that must not, or need not, be translated" in Chapter 5).

However, there are some things that can be done while developing the application that will help with these tasks, especially in the handling of hidden design elements and marking of text that will obviously not need translation.

Hidden elements

Hidden text does not need to be translated so you should make it easy for translators to find.

Always hidden text in forms and subforms

This is often a comment that explains what a hidden field does, appearing at the beginning or end of a form. To help translators, highlight the text using a different color in the form or subform and use the DO_NOT_TAG style to exclude it from the translation process. See "Excluding text from translation", below.

If you cannot use the DO_NOT_TAG style, mark the text in some other way, for example using the \\ convention, so that it can be quickly identified in glossaries or excluded from tagging altogether.

Indicate sections of hidden text with begin and end markers:

```
\\-----Begin Hidden Text-----\\  
\\-----End Hidden Text-----\\
```

Always hidden fields

These are fields whose content is used internally and never displayed to the user (not in other fields, views, etc). Precede the field name with an agreed symbol, for instance xx. The text associated with it can then easily be found and flagged 'Prevent Translation' by using an agent in the glossary.

Always hidden subforms

To help the translator, give hidden subforms a distinctively different color, for instance gray. Precede the subform name alias with an agreed symbol, for instance xx. Text associated with it can then easily be found and flagged 'Prevent Translation' by using an agent in the glossary. Insert the subform as a computed subform, by means of a formula. See "Subforms", below.

Hidden fields whose content is displayed elsewhere

You may want to include text in always hidden fields, and then display it somewhere else. This is a useful technique for text which appears in many different places in a form, and which you do not want to translate over and over again. Precede the hidden field name alias with an agreed symbol such as xD (for example, xDfl_Local). This way translators will recognize that it contains translatable text.

The displayable field should also be flagged with a prefix such as Disp (for example Disp_fl_Local). The displayable field name should also include part of the referenced field, in order to facilitate the connection.

Alternatively, if you allow WorkBench to create the tagged database with limited context matching, you can tag many instances of the same piece of text to one glossary entry.

Hidden views

There are two types of hidden view:

- Views which are used internally to perform lookups and data searches. These do not need to be translated. The view name should be written between parentheses and have the 'always hidden' prefix, for example (xx-AlwaysHiddenName). Text included in column titles should be preceded by a // symbol.
- Views which are accessed only via navigators or outlines. These do contain displayable text. Their names should include a suitable 'hidden but displayable' prefix, for example (xD-ViewMenuHiddenName).

Debugging and logging text

Make sure that you flag any text that has been included for debugging purposes. End users should not see this text so it does not need to be translated. Precede it with a suitable identifier, for example 'dbg_'. If you use it in LotusScript functions, name the function 'debug' or something similar. The same advice applies to internal logging functions which track what the application is doing. In such cases name the function 'logging'.

Include comments to explain wherever code has been added for debugging, and remove it when the application is stable.

'Shared / Personal on first use' folders or views

Any private folders or views that have been created by opening 'shared / personal on first use' folders or views should be removed from the database before localization because they will not be accessible in the language databases and to translate them would be a waste of effort. Users of the language databases will create their own private folders or views by opening the localized version of the 'shared / personal on first use' folder or view.

Excluding text from translation

If there are paragraphs in your database that you know will never need to be translated, for example comments about the design, you can make Domino Global WorkBench ignore them by giving them a paragraph style called DO_NOT_TAG. You can define the style in any way you like: it is the *name* of the style that the WorkBench looks for.

To assign the DO_NOT_TAG style:

1. In Domino Designer, open a design document that contains text you want to exclude from translation.
2. Put the cursor in the piece of text, right-click and choose Text Properties.
3. In the properties box, click the styles tab.
4. Click Create Style.
5. Enter DO_NOT_TAG, and check 'Make style available for all documents'.
6. Click OK.
7. Put the cursor on the next piece of text you want to exclude from translation, and select DO_NOT_TAG in the properties box.
8. Repeat Step 7 for other text that is to be excluded from translation.

There are other ways of excluding text from translation once the database has been designed. See "Terms that must not (or need not) be translated" in Chapter 5.

HTML

Domino Global WorkBench tags text in HTML provided it is handled in one of the following ways:

- It is marked with the pass-thru text style (to mark text in this way, highlight it and choose Text - Pass-Thru HTML).
- It is contained in a page where 'Treat contents as HTML' has been checked.

- It is stored in a field named 'HTML' (see "Storing HTML in a field" in the Domino Designer documentation). The default formula for such a field is tagged as HTML only if it comprises a single quoted string, otherwise it is tagged as plain text.
- It is stored in rich text fields which are flagged 'Treat contents as MIME/HTML'. The WorkBench tags text where the content-type is text/html.
- It is in a form that is marked 'Treat form contents as HTML'. The default formula for any field in such a form is tagged as HTML only if it comprises a single quoted string, otherwise it is tagged as plain text.

HTML tagging works best when the text that constitutes it is kept as plain as possible (the same applies to any JavaScript it contains). The presence of non-text Notes rich-text objects (for example fields, tables and buttons) or embedded subforms within an HTML stream make it difficult for the WorkBench to properly extract the translatable text. This is because when these objects are embedded, the HTML becomes fragmented into many smaller pieces which are difficult for the WorkBench to analyze.

For example, you should avoid embedding Notes fields or computed text in HTML:

```
<H1 color=0xFFCCFF>This is a heading</H1>
```

Here, the value 0xFFCCFF is held in a Notes field. If you need to construct dynamic HTML use formulas whenever possible, and inform the translators where they are.

Domino Global WorkBench does not support some elements embedded in an HTML stream, such as bitmaps. Fields and table markers are supported, but HTML that includes other embedded elements will have to be excluded from the tagging process (or marked 'Prevent Translation' in the glossary) and localized 'manually' in each output database, after building. Inform translators of any HTML in the database that contains embedded elements.

The following are processed as normal text (all text, including HTML coding, is tagged):

- HTML in paragraphs that have been given the paragraph style 'HTML'.
- HTML embedded in square brackets.
- HTML (or pieces of HTML) in formulas.
- HTML in a view or folder where 'Treat text as HTML' is checked.

Note If there is a design element property enabling HTML, that will take precedence over a pass-thru property of any type. So, for example, text in square brackets in a page marked 'Treat contents as HTML' is tagged.

Java

By default Domino Global WorkBench Release 5.0.5 tags text in Java .properties files for translation. See "Java" in Chapter 6.

Using existing terminology

If you already have Domino Global WorkBench glossaries that include translations, try to use terminology from them in your design. This will reduce the amount of translation work that needs to be done. Note that:

- Notes Global Designer Release 4.6 glossaries can be upgraded to Domino Global WorkBench glossaries. However, if you intend to use glossaries upgraded from Notes Global Designer Release 4.6, try to minimize the number of attribute changes in new text. See "Handling attributed text (rich text)" in Chapter 6.
- Terminology lists in the form of structured text files can be imported to Domino Global WorkBench glossaries.

See also "Upgrading an existing glossary" and "Importing" in Chapter 4.

Keeping translators informed

It is a good idea to keep translators informed of any aspects of your design that may have an impact on localization. You can do this by including comments specifically directed at translators. In particular:

- If there are elements of your design that are never displayed and also are not referenced by any displayed text that requires translation, inform the translators. This will save time, as these elements can then quickly be marked 'Prevent Translation' in the glossaries. Alternatively, you can exclude it from the translation process altogether by using the DO_NOT_TAG style as explained above.

Note Text that need not (or must not) be translated is often tagged anyway but marked 'Prevent Translation' in the glossary. This ensures it will always be recognized, and always treated as text to exclude from translation. It *could* be left untagged, but then, if the tagged database was rebuilt, it would not be found, and would be presented to the person doing the tagging as 'new' text again.

- If the functionality of your database depends on any pieces of text that are not aliased (for example file names or simple action references), inform the translators. Special precautions will need to be taken when tagging these, and some of them may have to be kept in the original language. Use REM statements in formulas to indicate such references. It will also help translators if you include REMs to explain the function of more complex formulas.
- If you have used 'hide when' formulas in your design, tell the translators where they are.

Design elements copied from other databases

If your design includes design elements that come from other databases, for example shared fields or subforms, make sure that the translators are aware of this, so that they do not translate them unnecessarily. One way to do this is to precede the aliases for the element names with a suitable prefix.

See "Subforms", below.

Preparing a database for building multilingually

When designing databases that will be built as multilingual databases, if you don't want every document to appear in all languages, you will need to include a language identifier field in each of your forms, and language selection as part of your view selection formulas, as explained in the following sections.

Documents created in a multilingual database

A document created with, say, a German version of a form can be identified programmatically as a German document *only* if it contains a suitable field that contains the right language identifier (de-DE). Domino Global WorkBench's workflow and synchronizing operations use a field called '\$Language' for this purpose. If you want to display documents according to language, you should therefore add this field to each form in your source database. Either give the field a fixed value, for example "en-US" if the source database is a US database, and then tag the identifier for translation. Or use a function such as @FormLanguage to give it a language value. For a description of @FormLanguage, see "Useful @functions", below.

There is one exception to this: if you add language synchronization features (see Chapter 8) the \$Language field, with the proper language value, is included automatically: it is on the subform that Domino Global WorkBench adds to synchronized forms.

Note You must enter the language identifier in the format shown here (two lowercase letters, a hyphen, and then two uppercase letters). The format used in glossaries is slightly different: the first two letters are uppercase and there is no hyphen. For a list of language identifiers, see "Adding new languages" in Chapter 4.

Navigators and views in a multilingual database

The navigators and views that the user sees are also selected automatically by Domino or Notes to be the ones with the appropriate language identifier.

However, the views will select documents in the same way as they did in the source database. To restrict views to documents of their own language, you must add language selection to their selection formulas. In each view in the source database, therefore, you would add \$Language="en-US" (if the source database was a US database). On localization, the string "en-US" must be translated in each language database (WorkBench does not tag view selection formulas). It would be translated "de-DE" in German. To select a language in general, rather than a specific variant, you might use @Left(\$Language,2)="en" in your selection formula.

If you wish a document to support multiple languages (for example global synchronized documents in a multilingual database) you will need to check that the language isn't contained in the language list so instead your formula should read @IsMember("en-US";\$Language).

By contrast, in a unilingual database the design is in one language only. Views will select documents just as they did in the source. In a German database (for example) users will see whatever has been created in this database (presumably German documents). To switch languages, a user simply looks at another database. In a set of Web site databases this will usually be done by clicking options on the home page (in Notes, doc links to the other databases).

About and Using documents (Help About and Help Using)

A Domino database, whether unilingual or multilingual, can contain only one About document and one Using document. If the source database includes either of these documents:

- When creating unilingual language databases, the WorkBench will build the appropriate translated documents.
- When creating multilingual language databases, the WorkBench will not build translated About or Using documents. Instead, it will copy the original documents, unchanged, to the multilingual database.

If you plan to create multilingual databases you can use Pages as localizable alternatives for About and Using documents (a database can contain more than one Page). The About and Using documents then need to include only a link to the relevant Page.

If you wish, you can make Notes display a specified Page when it opens the database. To do this, open the properties box for the database and choose 'Navigator' on the Launch tab. Finally select 'Page' as the navigator type.

Useful @functions

Notes Release 5 includes the following functions that are particularly useful in Domino Global WorkBench:

- @LanguagePreference([Content]) returns the 'Preferred Content Language' setting from the Notes client or the HTTP packet from a Web browser. The value returned is a string of two or five characters. The first two characters give the preferred language (for example 'en' for English) and the last two characters, if present, give the country variant (for example, 'US' for the United States). The letters returned may be upper- or lowercase.

For a list of language identifiers, see "Adding new languages" in Chapter 4. For information about setting the preferred language, see "Switching languages in built databases" in Chapter 7.

- @FormLanguage returns the language of the form used to compose the document. In databases built by Domino Global Workbench the value returned is a string of five characters of the form xx-YY, for example 'en-US' for United States English.

Often @FormLanguage is more useful than @LanguagePreference as it can only return a language supported by the database. For example, suppose someone is working in a multilingual database that supports French, German, Italian and Spanish, and suppose their primary language is Croatian, a language not supported by the database. Then, they will either see the database documents in the default language or, if they are using a browser and have set an available secondary language such as Italian, they will see documents in that language. Either way, if they compose documents whose \$Language field is given its value by @LanguagePreference, those documents will not be visible in any of the standard views because (usually) the views will select only French, German, Italian or Spanish documents. If, however, @FormLanguage is used instead of @LanguagePreference to give \$Language its value, the Croatian user's documents will display in the view they are able to see.

Limitations

The sections under "Good programming practice", above, explain how best to handle certain design features (for example formulas, LotusScript, JavaScript and HTML). Where appropriate these sections include information about limitations.

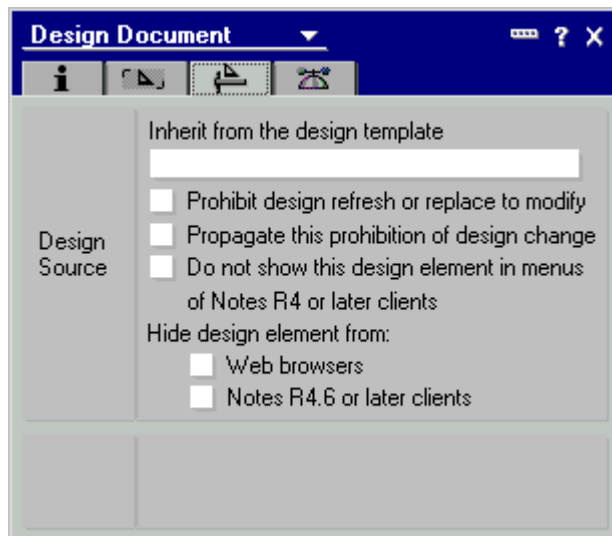
The following sections describe a few specific limitations that you should be aware of. Where possible, workarounds are suggested.

Attachments and embedded items

With the exception of Java files, the WorkBench does not touch attachments or embedded items, which must be localized separately and inserted manually into the built databases.

Design elements marked 'Prohibit refresh or replace'

The WorkBench will not process design elements marked 'Prohibit design refresh or replace to modify'. To access this setting select the design element in Domino Designer (do not open it), right click and choose Properties. Go to the third tab:



Make sure the first box is unchecked.

Caution The box is checked by default if you paste a design element into your application.

Help documents

Unlike other design elements, help documents (data documents) do not have any fixed title that the WorkBench can display; they contain whatever fields their designer chose to include on the parent form. Because of this, the WorkBench uses the views it finds in the database to display them.

The WorkBench also has a flat 'All Documents' display. This displays help documents as follows:

```
<NoteID> \ <Form> \ <Predefined field value>
```

where:

<NoteID> is the document's unique Notes ID.

<Form> is the name of the form used to create the document.

<Predefined field value> is the value of a field, if the WorkBench can find one that it recognizes. The WorkBench looks for the following common field names and displays the contents of the first one it finds: Topic, Subject, Author, Abstract, Title, Category, CompanyName, Product, Name, and Type.

Therefore, if you want the 'All Documents' display to be useful, make sure that a field with one of these names exists on any form used to create help documents. If necessary you can create a hidden field, and enter a formula to give it a suitable value. Typically the formula will simply be a reference to another field on the form.

Note Keep the number of help documents in your database as small as you can, because they slow down the WorkBench's update procedure, and remove any documents that are really data documents rather than help documents before work with Domino Global WorkBench begins.

Length limitations

Notes has certain limitations on the length of particular pieces of text (see below). This means that if the translated version of a piece of text exceeds the allowed length, there will be a truncation in the built database (in that language). Keep text for these items (especially navigator button captions) as short as you can.

Maximum lengths

Database title:	32 characters
Navigator button caption:	31 characters
View title:	63 characters*
View column headings:	80 characters
View comment:	255 characters
Form title:	64 characters
Field Help:	255 characters
Folder title:	80 characters
Folder column headings:	80 characters
Folder comment:	255 characters
SubForm title:	64 characters
Navigator title:	64 characters

* For multilingual applications the total length of translated cascaded view titles must not exceed 66 characters (including the backslash). If this limit is exceeded the application may crash without warning for any view accessed by db.getview() that follows after the view in question (including the view in question). For non-cascaded view titles the total length may not exceed 63 characters. Longer view titles will be truncated to 63 characters with the usual warning in the report database.

Referencing other design elements

In Notes, it is often possible to select from a list: for example if you are inserting an embedded view into a page you can pick the view you want from a list of views. However, for ease of localization it is better to avoid this method because the list displays the actual design element names rather than their aliases. Instead, check the 'Formula' radio button and reference the design element in a formula, usually just its alias between quotes:

```
"va_AuthorView"
```

\$\$ViewTemplate for, \$\$SearchTemplate for and \$\$NavigatorTemplate for

When creating these design elements, use aliases rather than actual names. For example, in a discussion template, instead of:

```
$$ViewTemplate for By Author
```

where By Author is the actual name of the view, you should use:

```
$$ViewTemplate for va_Author
```

where va_Author is the view's alias.

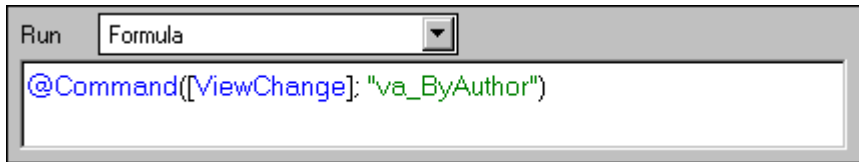
Simple actions

In Notes, it is often possible to reference other design elements through a 'simple action'. You choose a design element from a drop-down list. However, the list shows only the element *names*, not their aliases (if any), and this can cause problems when tagging (see "Tagging references to other design elements" in Chapter 6).

It helps if a reference like this:



Is changed to one like this:



In most cases, there is a workaround that consists in using an @Command or a formula instead of a simple action, specifying the alias of the design element as a parameter:

Simple action	Workaround
Form/view/agent simple actions	
Copy to Folder	@Command([Folder] ; "folder_alias")
Move to Folder	@Command([Folder] ; "folder_alias" ; 1)
Remove from Folder	@Command([RemoveFromFolder]) This is not an exact workaround, because you cannot specify the folder from which you want to remove the document, but it will work if the document has been opened from a folder (not a view).
Modify fields by form	Avoid using it. Use instead: FIELD field_name := "new_value" or Field := field_name; @SetField("field_name" ; "new_value")
Run Agent	@Command([ToolsRunMacro]; "agent_alias")
Send Newsletter Summary (only when using the 'Include summary for each document using view' option)	None
Navigator simple actions	
Open another Navigator	@Command([OpenNavigator] ; "navigator_alias")

Simple action	Workaround
Open a View	@Command([OpenView] ; "view_alias") or @Command([ViewChange] ; "view_alias")
Alias a Folder	@Command([ViewChange] ; "folder_alias")
Database launch options	
Open designated Navigator	None
Open designated Navigator in its own window	None

Subforms

There are two ways of using a subform in Notes:

1. You can embed it directly into the forms that need it.
2. You can insert it using a formula (a computed subform).

If you use the first method, *and you are working in a Notes 4.x environment*, the WorkBench will process the subform every time it finds it in a form. This can lead to a lot of unnecessary duplicated text in the glossaries, which in turn can lead to unnecessary translation work.

If you use the second method, however, the WorkBench will process the subform only once.

To help the translator, therefore, subforms should be inserted by means of a formula. The formula would simply be the name of the subform between quotation marks, for example:

```
"sa_SubFormName"
```

If, for performance reasons, you prefer to use embedded subforms, create always hidden text fields in your forms to indicate the start and end of each subform:

```
\\-----Begin Subform-----\\  
\\-----End Subform-----\\
```

The translators will then be able to locate text between the markers in the glossaries and mark it 'do not translate'. The subform itself should be translated in the usual way. Because the subform section of a form is refreshed each time the form is opened, the translation will automatically be used in place of the original text.

Note Notes Release 5 stores embedded subforms more or less like computed subforms. Therefore, the WorkBench does not tag them when they are embedded in a host form. So you do not need to flag the beginning or end of the inserted subform when working in a full Notes Release 5 environment.

See "Subforms" in Chapter 6.

Modifying source databases after localization has started

The WorkBench has updating capabilities that rely on the UNIDs of the design elements in the database. Therefore, if you need to modify the source database after localization has started, you should, wherever possible, edit existing design elements rather than create new ones. If a design element has been re-created (for example, by copying and pasting), rather than just edited, it will have a different UNID and the WorkBench will treat it as a new design element.

The WorkBench update procedure tags any 'new' design elements that have been created by copying and pasting existing ones, puts them into the tagged database, and from there builds them to the language databases. At the same time it removes their 'old' counterparts. However, the process is quicker if it can really *update* design elements rather than recreate them each time there is a design change.

Note To avoid creating new design elements, you can work on a copy or replica of the source database: UNIDs are preserved in copies and replicas, so the update procedure will work correctly. With previous versions of Domino Global WorkBench this was not possible, because NoteIDs were used for updating and these are not preserved in copies or replicas.

See "Updating" in Chapter 6.

Preparing an existing database

If you have an existing database application that needs to be localized, and you are not familiar with its design, you should at least consider adding aliases (preferably aliases with standard prefixes) where they do not already exist. This will allow the WorkBench to operate more efficiently and will speed the localization process. And if you can implement any of the other items discussed in the previous sections of this chapter so much the better.

Note If you want to build the database as a multilingual database, or you want to add language synchronization features (required for translation workflow), there are various elements that must be included in the original database design. These should be added by the database designer, or by someone who fully understands its design. For details, see "Preparing a database for building multilingually" in Chapter 3 and "Preparing a database for language synchronization" in Chapter 8.

Making a backup

First make a backup replica of the database. You can create it as follows:

- By using a standard Windows or DOS copy operation to copy the .NSF file.
- From inside Notes by choosing File - Replication - New Replica.

Note If you use the File - Database - New Copy command instead, the copy will be given a different replica ID and therefore will not be able to replicate with the original database or any of its existing replicas.

Creating a design synopsis

A design synopsis lists a database's design elements and the text they contain. You can use this to identify and mark design elements that might need changing:

- If you are able to change the design of the database, consider making changes along the lines suggested in "Tips for designers", above.
- If you are not able to change the design of the database, you should at least make sure that all views, forms and navigators are aliased. See below.

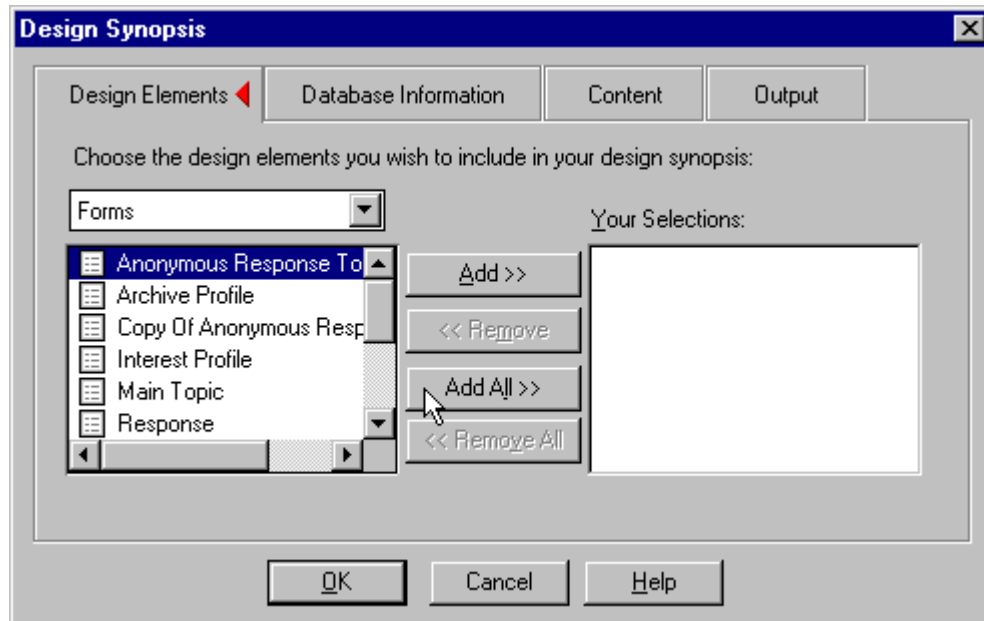
Note Keep the design synopsis. You can use it to identify the pieces of text in the database that need to be tagged (see "Tagging" in Chapter 6).

The procedure is:

1. Select the database icon on your Notes desktop.

Note If you can, use Notes Release 5 even if you are dealing with 4.6 templates or applications. The synopsis is more comprehensive and gives you access to all design elements, including LotusScript and JavaScript

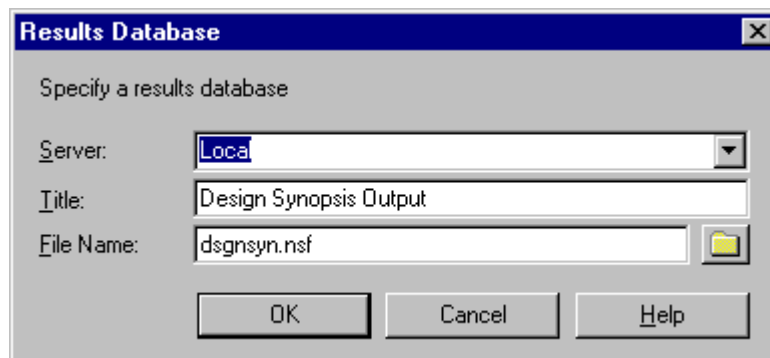
2. Choose File - Database - Design synopsis. Complete the dialog:



On the Design Elements panel, select each design type in the drop-down box and click Add All to select all elements of that type for the design synopsis. A design synopsis can be very long, so you may find it easier to produce it in stages, for example selecting forms first, then views, then agents, and so on.

If you want to reduce the amount of detail on the design synopsis, you can deselect items on the Content panel of the dialog.

3. On the Output panel, check 'Write Output to Database' and then click OK.
4. Select a database for the output:



If you are creating the design synopsis in stages (forms first, then views and so on) check 'Append to this database' for second and subsequent runs.

5. Click OK. Notes creates the design synopsis.
6. Repeat steps 1 to 5 for other design element types if necessary.

7. Open the design synopsis database in Notes. You will see the design synopsis document (or several documents if you created it in stages).
8. Open the design synopsis you want to review. You can print it (File - Print), save it (File - Export) and/or forward it to yourself. Because it can be long, you may want to review it in electronic form (and maybe reduce the font size) before printing. It cannot be edited directly.

Reviewing a design synopsis

After creating a design synopsis you should mark on it:

1. Any text that does not need to be translated. You can exclude this from processing by the WorkBench as explained in "Excluding text from translation", above.
2. Any cases where a design element has an alias, but a reference to it (for example in a formula) is using its display name, not the alias. In such cases you must either:
 - Change the reference to point to the alias
 - or
 - Make sure the design element name is not translated
3. Any of the following that have *not* been aliased:
 - Form names
 - Folder names
 - View names
 - Field keyword values, defaults, input validation and translation values
 - Navigator names

If an item has been aliased, you will either see an entry in an Alias field:

The screenshot shows a Lotus Notes Design Synopsis window titled "DESIGN SYNOPSIS for D:\PROJECT\Discuss1.nlf on server". Below the title bar, it says "Lotus Notes Database Synopsis - Generated at 09:54:00 on 31/07/98". The main content area is titled "Form Information" and contains a table with two columns: "Name:" and "Main Topic".

Name:	Main Topic
Alias:	fa_MainTopic
Last Modification:	31/07/98 09:53:03
Type:	Document
Window Title Formula:	StandardTitle := Subject + ("Response)"; "% Response Topic"; @WindowTitle = "Auth

Annotations on the right side of the table:

- A line points from the text "The name of the form (displayable text)" to the "Main Topic" header.
- A line points from the text "The alias of the form name (used for computation)" to the "fa_MainTopic" value in the Alias row.

Or you will see one or more vertical bars in the name:

View Information	
Name:	By Category va_ByCategory
Last Modification:	30/07/98 15:39:02
Default Database View:	No
No Response Hierarchy:	Disabled
Categories Initially:	Expanded
Unread Marks:	Compute and Display at All Levels
Refresh Frequency:	Automatic

The name of the view (displayable text)

The alias of the view name (used for computation)

Field:	fl_Status
Datatype:	Keywords
Help Description:	[Not Assigned]
Field Type:	Editable
Keyword User Interface:	Standard
Allow Values Not In List:	No
Allowable Keywords:	New ka_New Active ka_Active Complete ka_Complete Overdue ka_Overdue
Sign When Mailing/In Section:	No

Field keywords (displayable text)

The aliases for the keywords (used for computation)

Now create any missing aliases. The following section explains how to do this.

Creating or changing aliases

Using the design synopsis you marked in the previous section to help you, locate the design elements that have, or can have, aliases. Then:

1. Open the database in Notes.
2. Choose View - Design.
3. Select the first design element and open it.
4. Right-click the aliasable part to open its properties box.
5. Enter or modify the alias.
 - If you can, use an identifying prefix, as explained in "Aliases", above. If you do this you will have to track down all references to the design element in formulas and LotusScript, and change them to the new alias.
 - If you are not prepared to make changes to references to the design element, make the alias **exactly the same** as the original text. You can use Ctrl-C (copy) and Ctrl-V (paste) in property boxes to ensure text is the same.

Where there is no separate box for the alias, enter the text after (to the right of) the existing text, preceding it with a vertical bar character (|) as separator.

To view the values in a keyword field select the field and select Field properties in the properties box.

6. Close and save the design document.
7. If you used an alias prefix in Step 5 (or any other alias that is different from the original name), search the design synopsis for references to it and change them to the new alias.
8. Repeat Steps 3 to 7 for the other design elements.

Example -- aliased form name

The properties box for a form

Separator

This is the text that Notes will display

This is the alias (used for computation)

The screenshot shows the 'Form' properties dialog box. The 'Name' field contains 'Main Topic | fa_MainTopic'. The 'Comment' field is empty. The 'Type' dropdown is set to 'Document'. The 'Display' section has 'Include in Menu' and 'Include in Search Builder' checked. The 'Versions' section has 'Versioning' set to 'None'. The 'Options' section has 'Default database form' checked.

Example -- aliased keyword field values

Separators

The keywords (choices) that Notes will display

The aliases (used for computation)

The screenshot shows the 'Field' properties dialog box. The 'Display' section has 'Show field delimiters' checked. The 'Choices' section has a dropdown set to 'Enter choices (one per line)'. The text area contains 'New | ka_New', 'Active | ka_Active', 'Complete | ka_Complete', and 'Overdue | ka_Overdue'. The 'Options' section has 'Allow values not in list', 'Look up names as each character is entered', and 'Display entry helper button' checked.

Caution If your database design will be used to replace or refresh the design of an existing database, you should not make changes to keyword values because the existing keywords will already be used in database documents. Changing the keywords risks invalidating those documents.

Example -- aliased navigator name

The properties box for a navigator

Separator

This is the text that Notes will display

The alias (used for computation)

The screenshot shows a dialog box titled "Navigator" with a blue header and a grey body. The header contains a dropdown arrow, a menu icon, a help icon, and a close icon. Below the header is a toolbar with icons for separator, zoom, and grid. The main area contains the following fields and controls:

- Name:** A text field containing "Main Navigator | na_MainNavigator".
- Comment:** An empty text field.
- Initial view or folder:** A dropdown menu with the value "(\$All)".
- Auto adjust panes at runtime
- Web browser compatible
- Background color:** A color selection dropdown menu.

Example -- aliased view name

The properties box for a view

This is the text that Notes will display

The alias (used for computation)

The screenshot shows a dialog box titled "View" with a blue header and a grey body. The header contains a dropdown arrow, a menu icon, a help icon, and a close icon. Below the header is a toolbar with icons for separator, zoom, grid, and key. The main area contains the following fields and controls:

- Name:** A text field containing "By Category".
- Alias:** A text field containing "va_ByCategory".
- Comment:** An empty text field.
- Style:** A dropdown menu with the value "Standard Outline".

Chapter 4 Working with glossaries

For information about *translating* glossaries, see Chapter 5, "Translating".

Glossaries contain the text and translations used by the WorkBench to build localized databases. A Domino Global WorkBench glossary is a Domino database (.NSF file). A glossary contains:

- A term definition document for each piece of text.
- Translation documents associated with each term definition document. These are where translators enter translations.
- A glossary information document. This lists the active languages and the term types supported by the glossary. It optionally includes a description for the glossary.
- A locale information document for each language that can be supported by the glossary. See "Adding languages", below.
- Views that enable the contents of the glossary to be displayed in different ways.
- Agents to carry out management tasks, such as importing and exporting.

The 'All terms and translations' view in a Domino Global WorkBench glossary looks like this:

The screenshot displays the Lotus Notes interface for a glossary. The window title is "Glossary for X12 - All terms and translations - Lotus Notes". The menu bar includes File, Edit, View, Create, Actions, and Help. The toolbar contains icons for "New term", "Prevent translation", "Allow translation", and "Translate/Review". The left sidebar shows the "Domino Global WorkBench Glossary" with a "Views" section containing: All terms and translations, Available languages, Do Not Translate terms, Obsolete terms, Unapproved translations, Untranslated terms, Glossary information, and Languages. Below "Views" are "Actions" such as Export, Import, Retrieve..., Pseudo Translate, Selected Documents, All Documents, Delete Duplicates, Delete Orphan Translations, Delete Non-Linguistic Conflicts, and Help.

Term	Word count	Description
▼ X12 Discussion	5682	
▶ Agents	350	
▶ Folders	36	
▼ Forms	1623	
▼ Anonymous Response To R99		
All Documents	2	REM "variable"
English (United States) : All Documents		
French (France) : Tous documents		
German (Germany) : All Documents		
Italian (Italy) : All Documents		
Spanish (Mexico) : All Documents		
Anonymous	1	REM "Hhide if"
Anonymous Response To	4	
AnonymousResponsetoRe	1	
▼ Resource Images	112	
▼ AddTopic.gif AddTopic.gif	3	
action button		
Object: 810B6C1D	3	

The simplest way to add terms to a glossary is to let the WorkBench create them during tagging, as described in Chapters 6 and 7. However, you can add terms 'manually' from within Notes, and you can import terms and translations from structured text files (see below).

Graphics can also be stored in glossaries. Like text, they may need to be translated. See "Graphics", below.

Note For Domino Global WorkBench Release 5.0.5, the glossary design has been enhanced. To upgrade the design of an existing Domino Global WorkBench glossary, select it in Notes and choose File - Database - Refresh Design.

Release 4.6 issues

To use glossaries from Release 4.6 of Notes Global Designer with Release 5 of Domino Global WorkBench you must first upgrade them (see "Upgrading a glossary from Notes Global Designer 4.6", below). For information about building Notes 4.6 databases see "Using Domino Global WorkBench with Notes Release 4.6 databases" in Chapter 1.

The Domino Global WorkBench glossary application has been updated with Notes Release 5 design features which will not work with Notes Release 4.6. If you want to continue using Release 4.6 for translation work in glossaries follow these guidelines.

For an existing glossary (one used with a Domino Global WorkBench Release prior to 5.0.5):

1. Create a replica of the glossary.
2. Turn off replication of design for both replicas.
3. In the original glossary, change the template name from which it inherits design to '4.6 Compatible DGW Glossary', DGWGLOSS46.NTF.
4. Refresh the design of both replicas (File - Database - Refresh Design).

You can now use the original glossary for translation work with a Notes 4.6 client, and the new replica (the one created in Step 1) with this release of Domino Global WorkBench for tagging and building.




For a new glossary:

5. Create the glossary using the template DGWGLOSS.NTF.
6. Create a new replica of it.
7. In replication settings, turn off replication of design in both replicas of the glossary.
8. In the new replica you created in Step 2, delete the language views (all of the 'Translated/ Approved by Language' views).
9. In the new replica, change the template name from which it inherits design to '4.6 Compatible DGW Glossary', DGWGLOSS46.NTF. Refresh the design (File - Database - Refresh Design).

You can now use the new replica for translation work with a Notes 4.6 client, and the original glossary (the one created in Step 1) with this release of Domino Global WorkBench for tagging and building.

Note For operations such as importing terms to and exporting terms from the glossary, you should use the Release 5 replica which has improved functionality.

Local, Project and Global glossaries

When selecting a glossary in the WorkBench you can specify it as local () , project-wide () or global () . A local glossary is used for the current source database only; a project glossary is used for all source databases in the current project; a global database is used for all source databases in all projects.

Typically, a global glossary is a glossary of corporate terminology, used throughout corporate products and documentation. A project glossary contains standard terms for a particular field of activity. And a local glossary contains terms that are limited in use to the current source database. Global glossaries can also be used to ensure that a particular piece of text is translated in the same way (or left untranslated) throughout a project. This is because context matching does not operate when the WorkBench looks up terms in global glossaries.

You can remove a glossary from a source database by highlighting it and pressing the Delete key or right-clicking and choosing Remove From Project. If you remove a global glossary, it will be removed from all projects. If you remove a project-wide glossary it will be removed from all source databases in the project. If you want to remove a non-local glossary from just one source database, you must first change its type to local. To do this, right-click and choose Properties. Then select 'Local'.


Term definition documents

Each term in a glossary is defined in a term definition document. Associated with each term definition document are translation documents. Each translation document holds a piece of text (or a graphic) in one language: this is the text (or graphic) that the WorkBench uses when you build databases. When you create a new term, whether in Notes or by using the WorkBench, a new term definition document is created in the glossary database. At the same time a translation document is created for each of the active languages, ready for translation. A unique ID, which cannot be changed, is assigned to each term definition document when it is created.

Note Only the WorkBench can create term documents for graphics.


A term definition document has the following fields:

- On the Details tab:

Term format This can be Rich Text or Plain Text. See "Handling attributed text (rich text)" in Chapter 6. The field displays only when you create a new term document. Otherwise, an icon, , in the All Terms and Translations view indicates when a term has rich text format. You can change the format by clicking 'Promote to RTF' or 'Demote from RTF' in the term document.

Term in reference language This is the reference language version of the term. It can be a piece of text or a graphic. See "Reference language" and "Graphics", below.

- On the Settings tab:

Translation allowed By default this is set to Yes. If a term has been marked Do Not Translate you will see the lock icon () next to it in the Terms with Translations view. You can change this setting by clicking 'Allow Translation' or 'Prevent Translation' in the term document. See "Preventing the translation of terms in a glossary" in Chapter 5.

Term is obsolete By default this is set to No, but if a term becomes obsolete you can set it to Yes by clicking 'Mark Obsolete'. See "Marking terms as obsolete", below.

Term description This is an optional description of the term, typically a comment to help translators. When WorkBench creates a glossary entry from a piece of text in a formula, it puts the whole formula here so that translators can see the context of the text.

Term types These are the term types (if any) to which the term belongs. See "Term types", below.

- On the Context tab:

Context fields These indicate where the term came from, the design type and name and the design subtype and name. The WorkBench can use these for matching purposes when it creates a tagged database. The WorkBench fills out these fields automatically when it creates a glossary entry. The 'Archive', 'Property bundle file' and 'Resource key' contexts apply only to Java design elements.

Note The 'Check Consistency' button checks that all the necessary translation documents are present.

Translation documents

All the usable text and bitmaps in a glossary are held in translation documents. These are associated with the term definition documents that define the terms. Whenever a new term is added to a glossary, a translation document is created automatically for the term in each of the glossary's active languages.

Note You will also see a translation document in the reference language, because only translation documents are used as text sources by WorkBench. These documents are marked as Translated and Approved, and they cannot be edited.

A translation document has the following fields:

- On the Details tab:

Language The language of this translation document.

Original term (plain text) This is the text of the term in the reference language. Even when the term is a Rich Text term, the text is displayed here as plain text. If the term is a graphic, this text identifies the parent term document.

Term description This is an optional comment associated with the term. It comes from the original term definition document.

Translated term This is where you enter the translation of the original term. It can be a piece of text or a bitmap. If the term you are translating is a Rich Text term you can enter a rich text translation here. If the term is a graphic, a copy of the original image is placed here ready for translation. See "Translating a translation document directly" in Chapter 5.

- On the Translation Settings tab:

Translation allowed By default this is set to Yes. If a term has been marked Do Not Translate you will see the lock icon (🔒) next to it in the All Terms and Translations view, and this field will be set to No. The document cannot then be edited. You can change this setting by clicking 'Allow Translation' or 'Prevent Translation' in the term definition document.

Translation description This is an optional comment associated with the translation.

Term types These are the term types (if any) to which the term belongs. See Term types.

Translation status This can be Untranslated (🔍), Translated (✅), or Approved (👍). The relevant icons are shown against the translation documents in the All Terms and Translations view. Initially, the translation status is set to Untranslated. When a translator translates the term s/he changes the status to Translated. When a reviewer approves the translation s/he changes the status to Approved. Changing status is done by clicking action buttons rather than editing the translation status field directly.

- On the Advanced Options tab are two buttons which appear when the term is marked Translated or Approved:

Apply to Language Variants applies the translation to other translation documents *for this term*, provided their basic language is the same as the current one. For example, if the language of the current translation document is ESES, Spanish (Spain), the translation will be applied to ESAR, Spanish (Argentina), documents and ESMX, Spanish (Mexico), documents.

Apply to Same automatically applies the current translation to other terms that have the same original text (matching is case and accent sensitive). The translation is applied only to translation documents whose language exactly matches the current one. For example, if the language of the current translation document is ESES, Spanish (Spain), the translation will be applied only to ESES documents (provided the original text matches). Context is ignored.

See also "Using the Translate/Review dialog" in Chapter 5.

Reference language

The reference language of a glossary is the language of its term definition documents. Each term definition document defines a term, and associated with every term definition document is at least one translation document. A translation document contains the text for the term in one language. The translation documents hold the text that the WorkBench actually uses.

The reference language of a glossary is usually the language, or default language, of the project's source databases.

Planning

There are many different ways of setting up and using Domino Global WorkBench glossaries, for example you can:

- Put all terminology in a single glossary database, which is replicated worldwide. This is the simplest scenario, although size could be a problem.
- Put all terminology into a single glossary database, but use selective replication to make single-language replicas available at translation sites. The parent database might still become large, but the replicas will be much smaller. Size is important, for example, if translators are accessing glossary databases by modem.

This saves a lot of file management work, and allows many tasks to be centralized, for example, commenting, building and updating.

See "Replicating a glossary", at the end of this chapter.

- Put all terminology that is new for the current project into a new glossary database. Translation work can then be restricted to this new, small database, and the existing terminology database need not be touched.

This has the drawback that the translators will see new terminology in isolation.

- Divide terminology by application, having one glossary database for each application. Your databases will then be smaller and searching operations will be faster, which should speed up tagging and database creation and updating operations. However, there could be a lot of duplication of terms between glossaries. You might avoid this by having one glossary devoted to general, cross-application terminology. This option, with selective replicas as mentioned above, is the option favored by Lotus.
- Put all terms and graphics that must not be translated into a glossary of their own. The glossary will be used by the build process, but translators will never need to look at it.

Note There are other ways of dealing with text and graphics that must not be translated. For example, you can mark terms in a glossary so that translation is prevented (see Chapter 5). Or you can simply not tag the item (see "Tagging" in Chapter 6).

- Use term types to divide terms according to their usage. This might be by project or by subject matter. For example, you could have term types for financial, accounting and legal terms. See "Term Types", below.

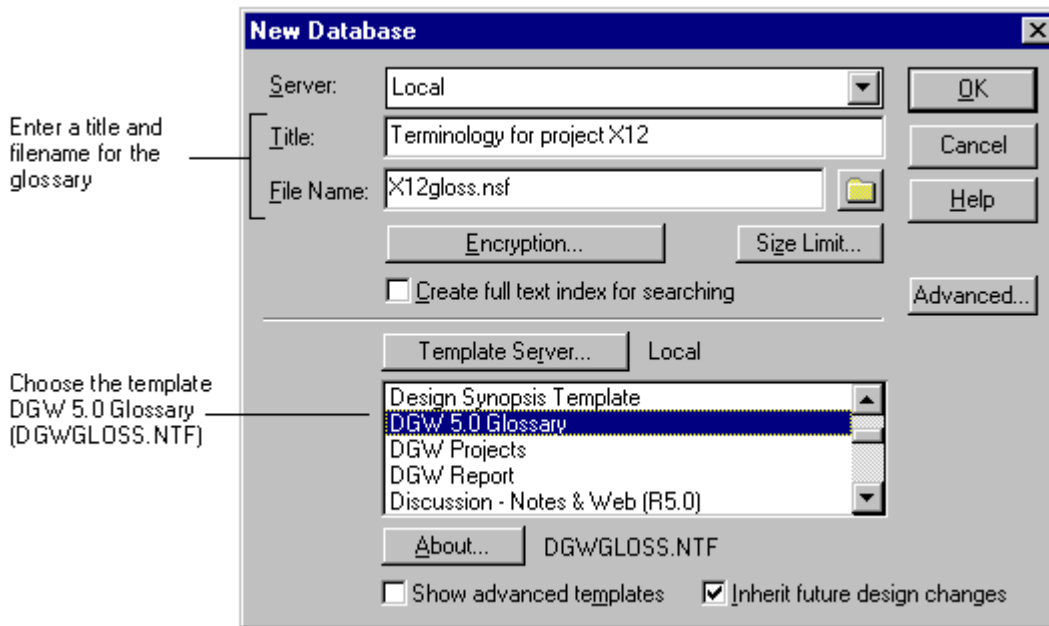
Creating a glossary

You can create a new glossary from Notes or from the WorkBench.

Creating in Notes

You create a glossary database as you would any other Domino database:

1. Start Notes.
2. Choose File - Database - New.
3. Complete the dialog box:



4. Click OK.

Caution Do not interrupt the glossary creation process. Scripts that initialize the glossary must be allowed to run to completion.

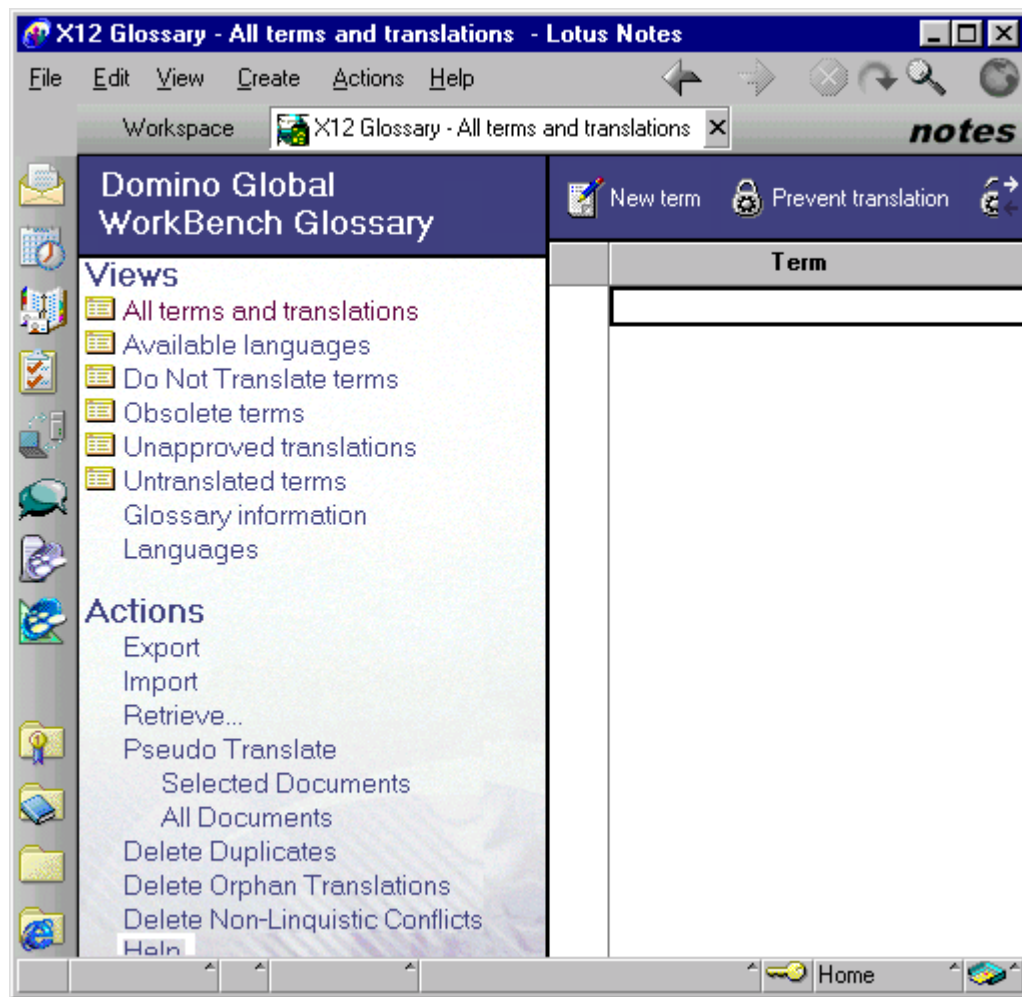
The glossary is created and you must select:

- (Optional) A descriptive comment for the glossary.
- The reference language. This will usually be the language of your source database(s).
- Language(s) to enable. These are the active languages (the languages that the glossary will support).

Tip It's a good idea to delay making languages active until you are ready to start translation because navigation in glossaries is faster with fewer documents. To begin with it is often advisable not to select any 'languages to enable': the reference language will be the only active language.

5. The Using document now opens.

6. Press Esc to close the Using document and you will see the main glossary display.



7. Add users to the access control list (ACL) and apply Translator and Approver roles as appropriate. See "Roles", below.

In order to ensure that you have a consistent set of glossary documents across languages, you might prefer not to give deletion rights to translators and approvers. Instead, make the deletion of terms a central role.

Creating from the WorkBench

You can create a new glossary as part of glossary selection:

1. With a source database loaded, choose File - Add/Create Glossary Database.
2. Click Create in the Select Glossary dialog.
3. Follow Steps 3 and 4 as for "Creating in Notes", above.
4. Back in the Select Glossary dialog, select a glossary type.

When adding a glossary you can specify it as local (📁), project-wide (📁), or global (🌐). A local glossary is used for the current source database only; a project glossary is used for all source databases in the current project; a global glossary is used for all source databases in all projects. The WorkBench consults them in that order: local first, global last.

5. Click Select to close the dialog box after adding a glossary. Click Add to keep the dialog box open so that you can add another glossary if you want to.

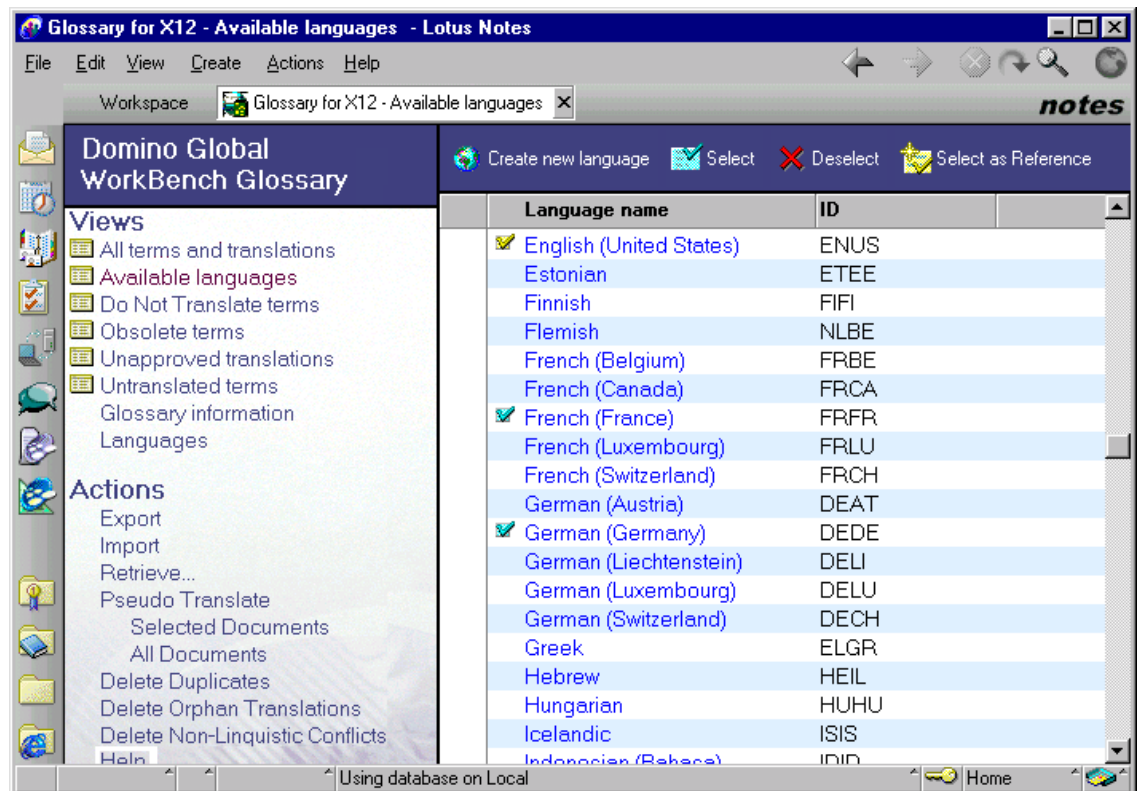
Making a language active

A glossary can accept translations only in its 'active' languages. The active languages are often, but not always, selected when the glossary is created. Before translation starts, check that all the languages that you want to support are active. If they are not, you must make them active. When you do this a translation document is created for every term in the glossary in each active language. Initially, data in 'Translated term' field of the translation document is copied from the 'Term in reference language' field of the corresponding term definition document.

Caution Making languages active in a glossary that has many terms in it can take a long time (you may need to let it run overnight), and it must not be interrupted.

To specify active languages:

1. Open the glossary database in Notes.
2. Click Available languages. You see the Available languages display. Languages that are already active are shown by a blue tick icon (✔). The reference language is shown by a yellow tick icon (☑).
3. Select one or more languages you want to make active and click Select. A new translation document, with status Untranslated (🔗), is now created for every term.
4. If the WorkBench is running, close it and then restart it in order to see the new active language(s).



Adding new languages

If you want to create translations in a language that is not listed in the glossary, you can define a new language. To do this:

1. Open the glossary database in Notes.
2. Click Available languages. You see the Available languages display.
3. Click - Create new language.
4. Edit the locale information document. You will have to enter:
 - The name of the new language.
 - The ISO 639 identifier for the language (this would be 'EN' for English).
 - The ISO 3166 identifier for the country variant (this would be 'US' for the USA). If this doesn't apply, repeat the entry you made above.
 - (Optional) A description for the new language.
5. Save the document.
6. If you plan to create synchronized databases that use the new language, open the database DGWSTORE.NSF in Notes (it is in the Domino Global WorkBench program directory). Create a language document for the new language (the simplest way is to copy and paste, and then edit, one of the existing documents).

Note The language ID in the glossary has no hyphen (for example 'ENUS') whereas in DGWSTORE.NSF it does have a hyphen (for example, 'en-US').

The following table lists the language and country identifier combinations that are currently supported by Domino Global WorkBench glossaries.

Code	Language/country	Code	Language/country
AFZA	Afrikaans	ETEE	Estonian
ARDZ	Arabic (Algeria)	EUES	Basque
ARBH	Arabic (Bahrain)	FIFI	Finnish
AREG	Arabic (Egypt)	FRBE	French (Belgium)
ARIL	Arabic (PNA)	FRCA	French (Canada)
ARJO	Arabic (Jordan)	FRCH	French (Switzerland)
ARKW	Arabic (Kuwait)	FRFR	French (France)
ARLB	Arabic (Lebanon)	FRLU	French (Luxembourg)
ARMA	Arabic (Morocco)	HEIL	Hebrew
AROM	Arabic (Oman)	HRHR	Croatian
ARQA	Arabic (Qatar)	HUHU	Hungarian
ARSA	Arabic (Saudi Arabia)	IDID	Indonesian
ARTN	Arabic (Tunisia)	ISIS	Icelandic
ARAE	Arabic (U.A.E.)	ITCH	Italian (Switzerland)
ARYE	Arabic (Yemen)	ITTT	Italian (Italy)
BEBY	Byelorussian	JAJP	Japanese
BGBG	Bulgarian	KOKR	Korean
CAES	Catalan	LAYU	Serbian (Latin)
CSCZ	Czech	LAXX	Latin
DADK	Danish	LOLA	Laotian (Laos)
DEAT	German (Austria)	LTLT	Lithuanian
DECH	German (Switzerland)	LVLV	Latvian
DEDE	German (Germany)	MKMD	Macedonian
DELI	German (Liechtenstein)	MSMY	Malaysian (Bahasa)

Code	Language/country	Code	Language/country
DELU	German (Luxembourg)	MTMT	Maltese
ELGR	Greek	NLBE	Flemish
ENAU	English (Australia)	NLNL	Dutch
ENCA	English (Canada)	NONO	Norwegian (Bokmal)
ENGB	English (UK)	NYNO	Norwegian (Nynorsk)
ENIE	English (Ireland)	PLPL	Polish
ENJM	English (Jamaica)	PTBR	Portuguese (Brazil)
ENNZ	English (New Zealand)	PTPT	Portuguese (Portugal)
ENUS	English (United States)	PSPS	'Pseudo-language'
ENZA	English (South Africa)	RMCH	Rhaeto-Romanic
ESAR	Spanish (Argentina)	ROMD	Moldavian
ESBO	Spanish (Bolivia)	RORO	Romanian
ESCL	Spanish (Chile)	RURU	Russian
ESCO	Spanish (Colombia)	SKSK	Slovakian
ESCR	Spanish (Costa Rica)	SLSI	Slovenian
ESDO	Spanish (Dominican Republic)	SQAL	Albanian
ESEC	Spanish (Ecuador)	SRYU	Serbian (Cyrillic)
ESES	Spanish (Spain)	SVSE	Swedish
ESGT	Spanish (Guatemala)	THTH	Thai
ESMX	Spanish (Mexico)	TRTR	Turkish
ESPA	Spanish (Panama)	VIVN	Vietnamese
ESPY	Spanish (Paraguay)	UKUA	Ukrainian
ESPE	Spanish (Peru)	ZHCN	Chinese (PRC)
ESPR	Spanish (Puerto Rico)	ZHHK	Chinese (Hong Kong)
ESUY	Spanish (Uruguay)	ZHSG	Chinese (Singapore)
ESVE	Spanish (Venezuela)	ZHTW	Chinese (Taiwan)

Making a language inactive

If you do not plan to translate into a language, you should not leave it active in the glossary. If you do, many unneeded documents will be created in your database.

Note You cannot make the reference language inactive.

To make a language inactive:

1. Open the glossary database in Notes.
If you already have translations in this language in the glossary, this process will remove them, so you may want to save them by exporting to a file. The process is described in "Exporting", below.
2. Click Available languages. You see the Available languages display. Languages that are already active are shown by a blue tick icon (✔). The reference language is shown by a yellow tick icon (👉).
3. Select one or more active languages you want to make inactive and click Deselect.

Caution All translation documents for the language(s) you are making inactive will now be deleted. The glossary will no longer accept translations in the language, and cannot build databases in the language.

Changing the reference language

You should not normally need to change the reference language of a glossary. However, you can change it to one of the currently active languages if you need to, provided that:

- The new reference language is currently an active language.
- The status of all the translation documents for the new reference language is Translated (👍) or Approved (👍👎), or there are no term definition documents yet in the glossary.

The procedure is:

1. Open the glossary database in Notes.
2. Click Available languages. You see the Available languages display. Languages that are active are shown by a blue tick icon (👍). The reference language is shown by a yellow tick icon (👉).
3. Highlight an active language that you want to make the reference language and click Select as Reference.

Note You can retrieve translations from another glossary (see "Retrieving translations from another glossary", below) only if its reference language is the same as that of the current glossary. The procedure described above provides a way of temporarily changing a glossary's reference language to allow retrieval in this way.

Term types

If you want to, you can assign terms to 'term types', which you define. You might want to set up term types for different projects (Project X, Project Y) or for different fields of use (word processing, communications, spreadsheets, databases, for example). Then you can use the term type to select for views or replication (the field name is Context_Category).

In Notes Global Designer Release 4.6 term types were called 'categories', and they were useful for pinpointing the context of a piece of text. Now, however, WorkBench automatically completes location (context) fields when it creates new glossary entries so you do not need to use term types for this purpose.

See "Location (context) fields", below.

Creating term types

You create term types as follows:

1. In Notes, open the glossary.
2. Click Glossary Information.
3. Click Edit Glossary Information.
4. Type the new term types into the 'Term Type(s)' field. Separate term types with a comma.
5. Save the document.

Note You can also create new term types when you create glossary entries, whether from the WorkBench or from Notes.

Using term types

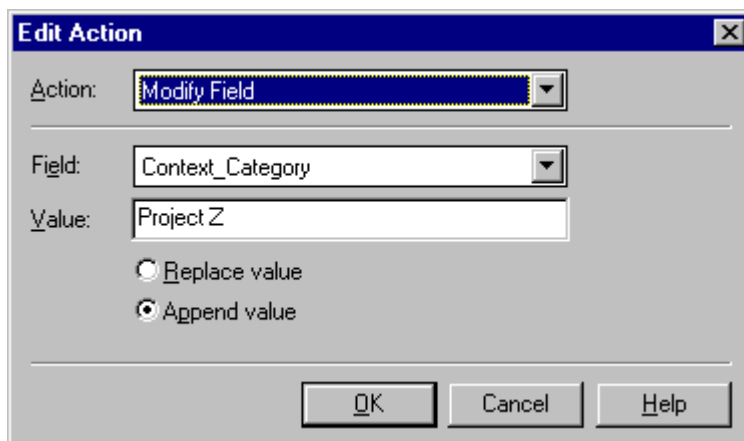
To assign a term to one or more term types:

1. In Notes, open the glossary that contains the term.
2. Highlight the term's definition document and click Edit.
3. Go to the 'Term type(s)' field, click the selector button and select one or more term types. You can add new term types if you want to.
4. Save the document.

Note When you create a new term from the WorkBench you can at the same time assign it to a single term type.

Assigning terms to term types quickly

The quickest way to assign terms to term types is to use an agent. You would use a simple action like this to assign selected terms to the term type 'Project Z' (already set up):



Note 'Append' value' is selected because some of the terms may already be assigned to other term types.

Reviewing a glossary before translation

Terms are added to glossaries either automatically by the WorkBench during tagging, or manually by entering them directly into the glossary database. Before translating the terms in the database you should:

1. Comment the glossary. Good commenting can have a very significant effect on the speed and quality of translation. See "Commenting glossary terms for translators" in Chapter 5.
2. Mark terms that must not be translated as 'do not translate'. See "Preventing the translation of terms in a glossary" in Chapter 5.
3. Make active the languages you are translating into (see above).

Tip It's a good idea to delay making languages active until you are ready to start translation because navigation in glossaries is faster with fewer documents.

Managing a glossary

The Domino Global WorkBench glossary template includes agents to help you carry out the various management tasks you may need to perform from time to time. The following sections describe these tasks.

Upgrading a glossary from Notes Global Designer 4.6

If you want to use glossaries that were prepared using Notes Global Designer Release 4.6, you must upgrade them before using them with Domino Global WorkBench Release 5.

Caution Upgrading a glossary that has many terms in it can take a long time (you may need to let it run overnight), and it must not be interrupted.

The process is:

1. Install Domino Global WorkBench Release 5 if it is not already installed.
2. Start Notes Release 5.
3. Create a copy (not a replica) of the existing glossary by using the Notes File - Database - New Copy command.
 - If you plan to use the glossary to build Notes Release 5 databases create the copy on a Release 5 server, or your local machine.
 - If you plan to use the glossary to build Notes Release 4.6 databases create the copy on a Release 4.6 server. See "Using Domino Global WorkBench with Notes Release 4.6 databases" in Chapter 1.

Caution Do not use the standard Windows or DOS copy operation, or the Notes File - Replication - New Replica command. These create replicas, not a copies.

Note If you want to use the glossary to build both Release 5 and Release 4.6 databases you should create two copies of the glossary.

4. Select the glossary copy's icon on your desktop.
5. Make sure that you have 'Manager' access rights (File - Database - Access Control).
6. Choose File - Database - Replace Design.
7. Select DGWGLOSS.NTF as the design template and click Replace.
8. Open the database. You will see a message "Glossary not fully updated. This operation may take some time."
9. Click OK to complete the procedure.

Domino Global WorkBench glossaries cannot be used with Notes Global Designer Release 4.6.

Roles

A Domino Global WorkBench glossary has two restrictive roles to which you can assign users:

- **Translator** Users assigned as translators can edit and save translation documents, but they cannot create new terms and have no access to management or review functions. Only the appropriate action buttons appear in the glossary display.
- **Approver** Users assigned as approvers have the same rights as translators, but can also mark translations as approved.

Note Users not assigned to either role have access to all the functions of the glossary. The roles are actually named 'rTranslator' and 'rApprover' in the glossary template, as an aid for localization groups.

If you assign users to these roles, you must implement a consistent access control list across all replicas of the glossary database. To do this:

1. In Notes, select the glossary database icon.
2. Choose File - Database - Access Control.
3. Click Advanced.
4. Check 'Enforce a consistent Access Control List across all replicas of this database'.

Adding categorization to glossary views

If you would like to see untranslated documents categorized by design element, as in the All Terms and Translations view, do the following (this method can be used to add categorization to any of the language views):

1. Create a new view:
 - Create a copy of the All Terms and Translations view.
 - Turn off the 'Show response documents in a hierarchy' setting in the new view and replace its selection formula with the selection formula from the Untranslated Translations view.
 - Copy the first (Language) column from the Untranslated Translations view and insert it as the first column of the new view.

The new view should now display a number of 'Uncategorized' categories. The agent below will fill these in.

2. Create an agent:
 - The agent should be set to run on selected documents.
 - Enter the formula:



```
FIELD Context_Database := @GetDocField($Ref;"Context_Database");
FIELD Context_Element_Name := @GetDocField($Ref;"Context_Element_Name");
FIELD Context_Element_Type := @GetDocField($Ref;"Context_Element_Type");
FIELD Context_Sub_Name := @GetDocField($Ref;"Context_Sub_Name");
FIELD Context_Sub_Type := @GetDocField($Ref;"Context_Sub_Type");
```
3. Run the agent:
 - Select all documents in the view.
 - Run the agent. The proper categories desired should now appear.

Notes There will be a noticeable increase in the size of the database because of the new view and additional items in every document. Once the agent has been run, you can simply add the category fields from the All Terms and Translations view to any of the language views if you want categorization in these views also.

Creating new terms

New terms in glossaries are generally created during the tagging process by the WorkBench. However, you can also use Notes to create new terms for text (but not for bitmaps). The procedure is:

1. In Notes, open the glossary.
2. In the All Terms and Translations view, click New Term.

3. Complete the Term form (see the illustration below):
 - If appropriate, choose one or more term types, see "Term Types", above.
 - (Optional) Complete the context fields (see below). These will pinpoint the context of the new term. During tagging, the WorkBench can use these fields for matching (such matching is case sensitive).
4. Click Save to save the new term definition document. When you save, a translation document is automatically created for each active language in the glossary, ready for translation. Until a translation is provided, these translation documents have the status Untranslated, and are marked with a question mark icon, . See Chapter 5, "Translation".
5. (Optional) Choose View - Refresh (F9) to update your view, or click the refresh icon, . You will see all the translation documents for the term in the view.

If the text contains attribute changes you may want to select 'Rich Text'. See "Handling rich text"

The text for the new term, in the glossary's reference language

Add a comment if necessary (for example, additional context information) *

Select a term type if necessary (see "Term types", above)

Enter information about the context of the new term *

* These are displayed in the WorkBench's Lookup dialogs

Context (location) fields

When terms are created by the WorkBench these fields are filled in automatically. They indicate where the term came from, from the broadest context (database name) down to the narrowest (design subtype title). See the table below.

When tagging a database, the WorkBench can match against one or more of these fields when it searches for text in glossaries (see "Creating the tagged database" in Chapter 6). So, for example, you can make the WorkBench tag new text to existing glossary entries only if the latter originally came from elements of the same type (for example, form or view) in a database with the same name.

Note When looking up text in *global* glossaries, the WorkBench *never* pays attention to context.


Context fields provide valuable context information for translators.

Context field	Value
Database name	The name of the source database.
Type	The following type labels are used by the WorkBench: Forms, Views, Folders, Shared Fields, Shared Action, SubForms, Navigators, Agents, "About Database" Document, "Using Database" Document, Icon, Database Title, Script Library, Database Library, Help Document, Template Name, Database Class, Outline, Page, FrameSet, Resource Images, Resource Java and Unknown Design.
Type title	The name of the design element, including its alias if it has one.
Sub type	The following sub-type labels are used by the WorkBench: Global objects: Database Title, Template Name, Database Class, Form Formula, View Selection Formula, Agent Formula, Window Title, View Comment, Default Navigator, Default Web Navigator, Launch Web View Name, Agent Comment, Subform List, Title, Column Title, and Column Formula. General: HTML Field Attribute, Pass-Thru HTML, LotusScript Text, Static Rich Text, Field (for help documents), Bitmap (for bitmaps in forms and navigators), and Alternate Text. In forms: Static Label, Field Default Value Formula, Field Input Value Formula, Field Input Validation Formula, Field Description, Field Keyword Formula, Field Keyword Value, Hotspot, Hotspot Popup, Hotspot Section, Hotspot Button, Hotspot Action, Hotspot Link, Hotspot Computed, Hotspot Script, Hotspot Button Script, Text Section Title, Formula Section Title, Button Text, Hide Paragraph when formula, DocLink Comment, Name Anchor, Header/Footer, Embedded View Name, Embedded Folder Name, and Embedded Navigator Name. In navigators: Navigator Text Record, Navigator Initial View/Folder, Navigator Button Label, Navigator Polygon Label, Navigator Rectangle Label, Navigator Round Rectangle Label, Navigator Ellipse label, and View Map Action Script. In simple actions: Simple Action Modify by Field New Value, Simple Action Reply Body, Simple Action Formula, Simple Action Sendmail To, Simple Action Sendmail Cc, Simple Action Sendmail Bcc, Simple Action Sendmail Subject, Simple Action Sendmail Body, Simple Action By Form, Simple Action Move To Folder/Copy to Folder/Remove From Folder, Simple Action News Letter View Name, Simple Action News Letter To, Simple Action News Letter Cc, Simple Action News Letter Bcc, Simple Action News Letter Subject, Simple Action News Letter Body, Simple Action Run Agent, Simple Action Run Script, Simple Action Search Query, Simple Action Query By Field, Simple Action Query Formula, Simple Action Query By Form, Simple Action Query By Folder, Simple Action Query Uses Form, Action Title, Hide Action Formulae, Action Open View, Action Open Navigator, and Action Alias Folder. Event Formulas: QueryOpen, PostOpen, QueryClose, QueryModeChange, PostModeChange, PostRecalc, QuerySave, RegionDoubleClick, QueryOpenDocument, QueryAddToFolder, QueryPaste, PostPaste, QueryDragDrop, PostDragDrop, PostDocumentDelete, QueryDocumentDelete, and QueryDocumentUndelete.
Sub type title	The name, if any, of the design sub-element, including its alias if it has one.
Archive	For Java sources, the name of the archive file (.JAR or .ZIP) that contains the text. If the resource file is held in a Notes design element and there is no archive file this context field is either blank or contains the text 'N/A', but the other context fields (above) allow you to identify the design element in question.
Property Bundle File	For Java sources, the name of the resource file (.properties) that contains the text.
Resource Key	For Java sources, the key that contains the text.

The Prevent/Allow translation buttons

Sometimes it is important that a piece of text in a database should remain in the reference language (see "Terms that must not, or need not, be translated", in Chapter 5). If you do not want a term or a graphic to be translated, select it in the 'All terms and translations view' and click 'Prevent translation'. It will then not be possible to edit the translation documents. To reverse the process, click 'Allow translation'.

Rich text terms

Domino Global WorkBench allows you to enter glossary terms in rich text format. This is useful where text includes attribute changes (for example font changes or words in bold or italics). Rich text entries are indicated in a glossary by an icon, .

Certain glossary operations cannot be performed with rich text entries. In particular:

- Rich text terms can be exported only as plain text.
- Rich text terms cannot be imported.
- The Translate/Review form cannot be used with rich text terms.
- The retrieve translations function cannot be used for rich text terms.
- The delete duplicates function cannot be used for rich text terms.


See "Handling attributed text (rich text)" in Chapter 6.

Changing to or from rich text format

You can change the format type of a term in a glossary as follows:

1. Open the glossary in Notes and select Terms with Translations view.
2. Open the term definition document of the term you want to change.
3. Click Promote to RTF or Demote from RTF.
 - If you are changing from RTF to plain text, all the attributes in the term's text (and its translations) will be removed, and its format will be set to Plain text.
 - If you are changing to RTF from plain text, the format will be set to RTF. If you make any changes to the text, or its attributes, the text in any translation documents will revert to the reference language text.
4. Save the document.

Graphics

As well as storing text, Domino Global WorkBench glossaries can store graphics that appear in design elements, and their translations. The procedures are exactly the same as for text terms. The WorkBench creates graphics entries in glossaries as it creates the tagged database (provided you have selected graphics for tagging), just as it does for text. A graphic entry can be marked 'Prevent Translation' if necessary. The translation of graphics is done by importing translated versions of the graphic to the associated translation documents. Graphic entries are held in standard term definition documents as rich text items, and they are indicated by an icon, .

Caution Notes Release 5 stores JPEG and GIF images in their native formats, whereas Notes 4.6 (and earlier releases) stores all graphics in BMP format. Therefore you must make sure that you do not use a Notes 5 client to translate graphics for use in Notes 4.6 compatible localized databases.

If you want to prevent the tagging of graphics, deselect them in the Tag the Following list box on the Tagging Options dialog.

See "Graphics" in Chapter 3 for some general guidelines on using graphics.

Importing, exporting and retrieving terms and translations

You can import terms from, and export them to, a structured text file. And you can retrieve translations from another glossary database.

Structured text files

A structured text file is a text file that contains data organized in columns (fields) and rows (records). Each record is a line in the file, and contains a different set of information. Each field is separated from the next by a standard character (a delimiter), such as a comma or a vertical bar (|). A simple terminology file of, say, 100 terms would contain 100 lines (records). On each line there would be first the reference language text, then the field delimiter, and then the translated text.

In the case of term types (a term can belong to several term types) you must separate each term type by a secondary delimiter character, which must be different from the main one.

When importing to a glossary you must generally specify the structure of your text file on the import dialog box. This indicates which delimited field in the file contains what information. Each line in the file is imported as a separate term in the glossary, and each piece of delimited information in the line is mapped to a field in one or other of the term's documents.

Note It is safest when creating structured text files to put each field entry between double quotes. In the case of term types put each separate term type between quotes.

Importing

Note Import can only import to, or create, terms whose text format is Plain Text. It cannot import rich text terms. See "Handling attributed text (rich text)" in Chapter 6.

The procedure is:

1. Check that the character set used by the terms you are importing is supported by the operating system and the Notes software you are running. If you need to change the character set in Notes:
 - Choose File - Tools - User Preferences and select the International panel.
 - Click Import/Export Character Set.
 - Select the character set of the file being imported.
 - Click OK.
2. If you want to add the terms to a new glossary, first create the glossary.
3. Open the glossary in Notes, and click Import.
4. Choose the Import format on the Import dialog box. This can be DGW Format (the file was originally exported from this glossary, see "Exporting", below) or Configurable.

If you chose DGW Format you specify only the name of the structured text file, *which must have been created from this database*, and (optionally) a log file.

If you chose Configurable you must specify:

- The name of the structured text file and (optionally) a log file.

Note The structured text file *must* include the text in the reference language for each term being imported.

- The language of the terms you are importing (it must be an active language).

- Import options: whether or not new terms may be created in the glossary; whether or not text in the structured text file is enclosed in double quotes; and whether to apply imported information to all terms with matching reference text, or just the first match (in the All Terms with Translations view).
- The order and contents of the fields in the structured text file. You do this by selecting parameters in drop-down boxes. If you choose Ignore, the contents of that field in the structured text file will be ignored. A graphical representation of the selected structure is shown at the bottom of the dialog box (an xx indicates a field to be ignored).
- The delimiters used in the file.

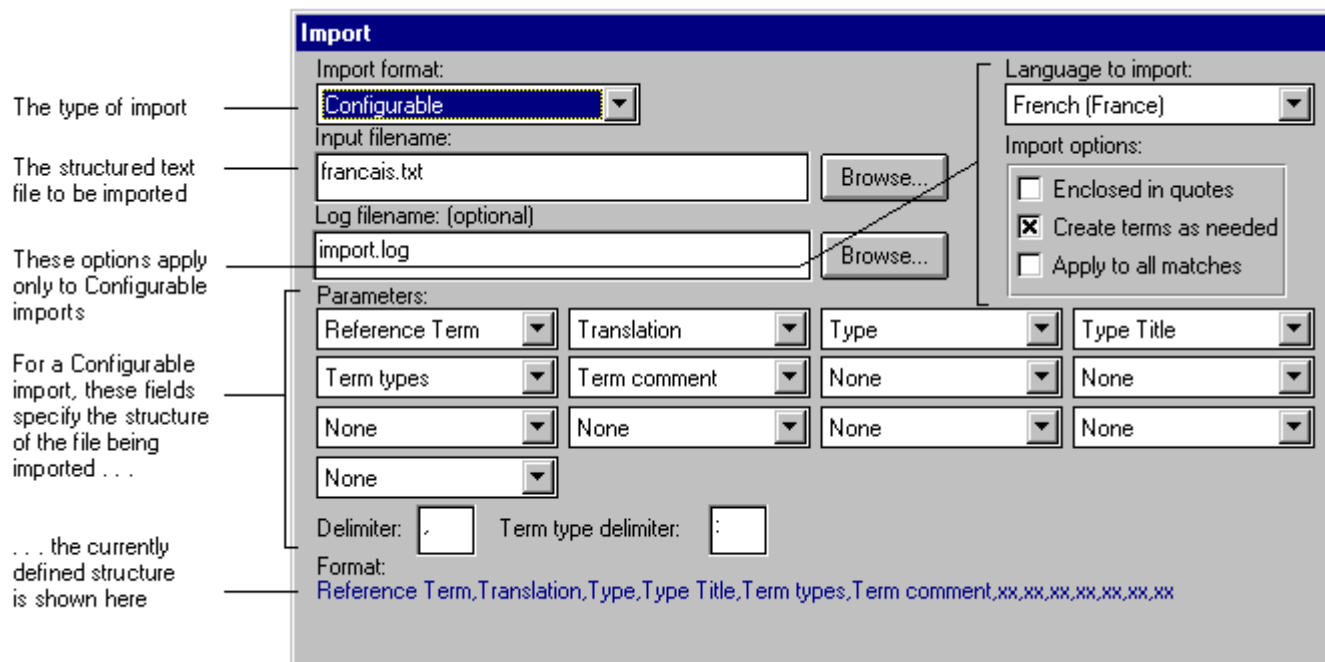
Note It is important that the delimiter characters are used *only* as delimiters, that they are not found in any of the terms you are importing.

5. Click OK. The information from the structured text file is imported.

If you chose Configurable:

- If a match is found with a term in the glossary (the reference language text in the structured text file matches the reference language text in a glossary entry), the information in the structured text file imported to the glossary document(s), overwriting any existing contents of the relevant fields.
- If no match is found in the glossary, depending on what you have selected on the Import dialog box, a new term may be created for the imported information.

Note Prevent Translation (DNT) settings are not imported to terms that already exist in the glossary. They are imported only when new terms are created in the glossary.



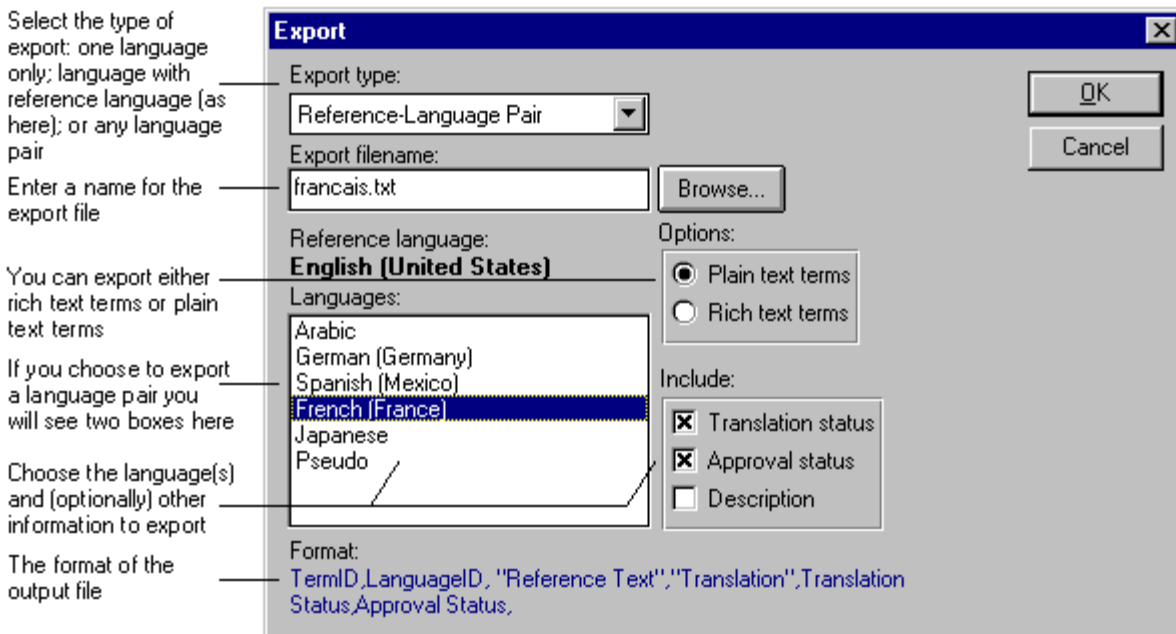
Exporting

Exporting text from a glossary can be useful if you want to translate terms outside the glossary database or run a standard procedure, for example a spelling check.

Note Export can export text only as plain text, even when it comes from glossary entries whose format is Rich Text. See "Handling attributed text (rich text)" in Chapter 6.

The procedure is:

1. Open the glossary in Notes, and click Export.
2. Complete the Export dialog box:
 - Select the language to export.
 - Select the export type. You can export both reference language text and translation language text, just translation language text, or text in any pair of languages.
 - Enter a filename for the exported file.
 - Choose whether to export text from plain text terms or rich text terms (see the Note, above).
 - (Optional) Choose additional information to include in the export: 'Translation status', 'Approval status', and/or 'Translation description'.
3. Click Export.



Retrieving translations from another glossary

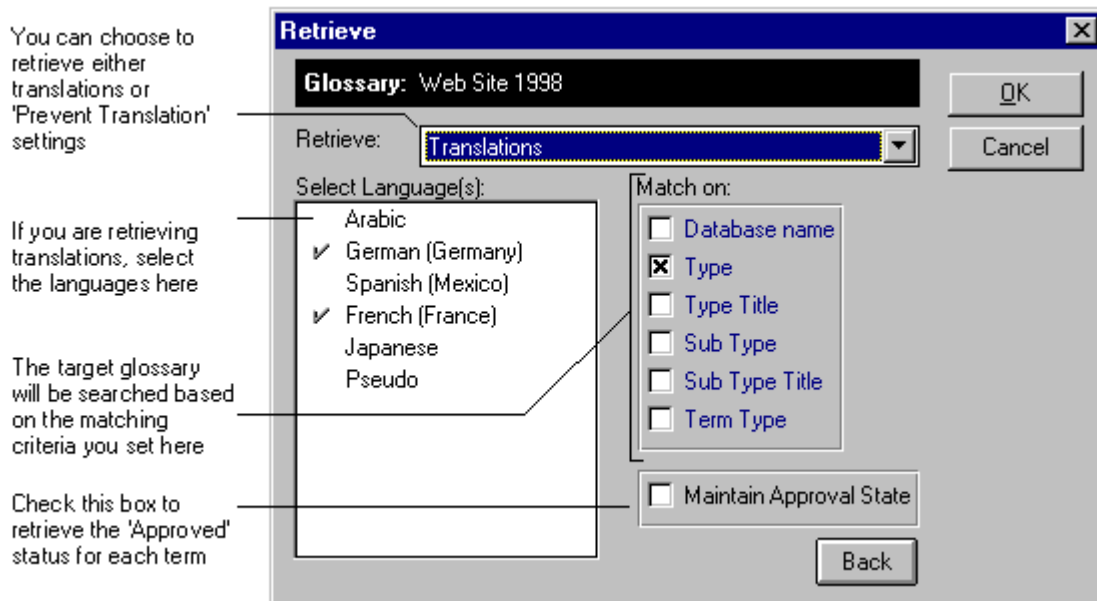
The retrieve procedure allows you to retrieve translations, or Prevent Translation (DNT) settings, from another glossary, for example a glossary that is being used for a different project, or one that was used for a previous version of the current project.

Note The glossary you retrieve from must have the same reference language as the one you retrieve into. You cannot retrieve rich text terms from another glossary.

The procedure is:

1. In Notes, open the host glossary (the one you want to retrieve translations into).
2. Click Retrieve.
3. Select the glossary from which you want to retrieve the translations and click OK. If the glossary is not listed you can enter its pathname directly, or click the Browse button.

4. Complete the Retrieve dialog box:
 - Select whether you want to retrieve translations or Prevent Translation (DNT) settings.
 - If you are retrieving translations, select the language(s), and whether or not to retrieve their Approved settings.
 - (Optional) Select context matching criteria (Domino Global WorkBench will always use the reference language text).
5. Click OK. Then, Domino Global WorkBench compares the reference text and context entries (if you have specified any) of each term in the host glossary with the terms in the glossary you are retrieving from. When it finds a match (if there are several matches, the first is used):
 - If you are retrieving translations, it copies the text from the specified translation documents into the host glossary. Text is retrieved *only* if it has status Translated or Approved, and *only* if the text in the host glossary has status Untranslated.
 - or
 - If you are retrieving Prevent Translation (DNT) settings, and the Prevent Translation setting of the found term is Yes, it changes the Prevent Translation setting in the host glossary to Yes.




Deleting terms

To delete a term from a glossary:

1. In Notes, open the glossary.
2. Select the term definition document(s) of the term(s) to be deleted.
3. Click Delete Terms. The term, and all its associated translation documents will be deleted.

Caution Once deleted, the term and its translation documents cannot be recovered.

4. If you see the refresh icon, , click it to update the glossary's view (or press Shift-F9 to rebuild the view index).


Notes You cannot selectively delete translation documents: every term in a glossary must have a translation document for every active language. You can, however, delete all the translation documents for a particular language by making that language inactive. See "Making a language inactive", above.

If, after deleting a term, you want to make the WorkBench re-tag to a new term without redoing the whole tagging process, you can force an update. See "Forcing an update" in Chapter 6.

Locating and marking obsolete terms

If, during tagging, you check the 'Mark terms' box in the Tagging Options dialog, the text in the 'Marker text' box will be added to every glossary term that is used during the tagging process. This allows you to identify quickly any glossary entries that are obsolete. By default the marker text is today's date. After tagging, any glossary entry that does not have the marker text in its Fld_touch field was not used when creating the tagged database, and will therefore not be used when building the language database(s). To view glossary entries by marker text, set up a view that uses the field Fld_touch as one of its categories.

Once you have located them, you can mark term(s) as obsolete in the glossary. The procedure is:

1. In Notes, open the glossary.
2. Select the term definition document(s) of the term(s) to be marked as obsolete.
3. Click Mark Obsolete. The term's 'Term is obsolete' field will be set to Yes.
4. If you see the refresh icon, , click it to update the glossary's view (or press Shift-F9 to rebuild the view index).

Obsolete terms remain in the glossary until such time as you delete them, but they cannot be used by the WorkBench for tagging or building. They (and their translation documents) can be viewed only in the Obsolete terms view. You can make a term current again by selecting it in this view and clicking Mark Current.

Housekeeping tasks

There are a number of tasks that you may have to carry out from time to time.

Deleting duplicates

This procedure can delete two sorts of duplicates in glossaries:

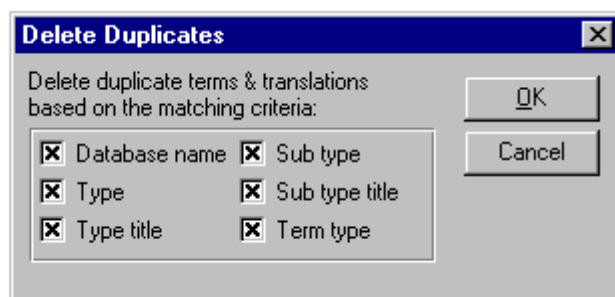
- Exact duplicates. These can arise when you create a tagged database, selecting 'Automatically for Each New Term' on the Prompting panel of the Tagging Options dialog box.
- Context duplicates. These are entries that have the same text, but differ from each other in one or more of their context fields.

The procedure is:

1. In Notes, open the glossary.
2. Click Delete Duplicates.
3. Complete the Delete Duplicates dialog box.

Note If you make the matching criteria less than exact, more than one match may be found, and the matching terms may differ from each other. All but the first (in the All Terms with Translations view) will be deleted.

Duplicates are terms whose reference language text, and any contexts checked here, are the same



4. Click Yes.

Deleting orphan translations

'Orphan' translations are ones whose term definition document is missing. Such translation documents can cause problems for the WorkBench during tagging and building. To remove them:

1. In Notes, open the glossary.
2. Click Delete Orphan Translations.

Orphan translation documents can arise if a term is deleted from a glossary replica that contains only a subset of the supported languages. Translation documents for the languages not in the subset replica will then remain in one or more of the other replica(s).

Deleting non-linguistic conflicts

Sometimes a replication conflict can result in two translation documents for the same term and the same language. In cases where one has status Translated or Approved and the other has status Untranslated, you can delete the latter automatically as follows:

1. In Notes, open the glossary.
2. Click Delete Non-Linguistic Conflicts.

Note You should carry out this procedure in one replica only, allowing the deletions to pass to other replicas by the normal replication process. Running the process in more than one replica is likely to create more replication conflicts.

Deleting other replication conflicts

If a translation document is edited in two different replicas between replications you will see a replication conflict. To delete one of the conflicting translation documents, highlight it and press the Delete key.

Creating 'pseudo-translations'

Pseudo-translation is a way of seeing some of the effects of translation, for example on the functionality and layout of your application, before translation proper is done (it is more efficient to make adjustments *once*, in the source or the glossary, rather than in every built language database or resource file).

If the language of your source database is English, you will find that translations are up to 30% longer. This can lead to some problems with layout and truncation (see Chapter 7, "Using built databases"). Of course if your development language is a 'longer' language, such as German, you will be unlikely to encounter expansion problems.

Pseudo-translation creates a set of translation documents automatically in a glossary, giving them the 'language' Pseudo. The basic text for these documents is the reference language text, but it is transformed in a way you specify:

- It can be expanded by a percentage you specify. After building this will show up layout and truncation problems.
- It can be reversed. After building this will show up text that is not being 'translated': such text will appear normally (not reversed) in the reference language. Any untranslated text has either not been tagged, or it has been marked 'Prevent translation' (DNT) in the glossary.
- It can be expanded and reversed.

Note You can run the process only *once* for any given term.

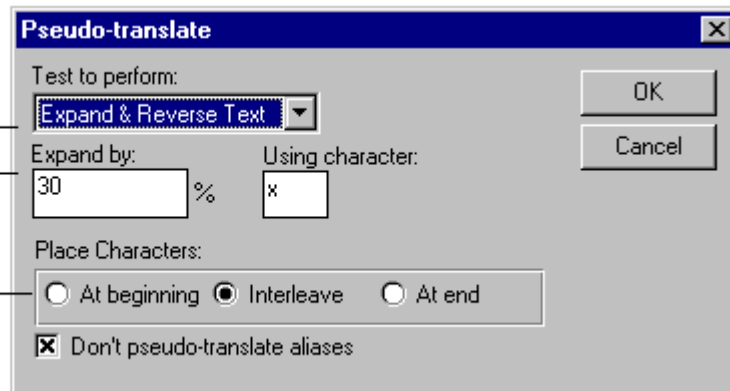
The procedure for creating pseudo-translation documents in a glossary is:

1. In Notes, open the glossary.
2. If Pseudo is not already an active language, make it active.
3. If you want to create pseudo-translations for selected terms only, select the terms now.
4. If you made a selection in Step 3, click Pseudo Translate / Selected Documents. Otherwise click Pseudo Translate / All Documents. The pseudo-translation dialog box opens.
5. Complete the dialog box and click OK.

You can create pseudo-language translations by expanding and/or reversing the reference language text

Expansion will be by the amount specified here, using the character specified

Added characters can go at the beginning or end of the text, or within it



Note Normally you should leave 'Do not pseudo-translate aliases' checked.

After creating pseudo translations, you should build a language database, specifying 'Pseudo' as the build language. See "Building" in Chapter 6.

Now you can test the built pseudo-language output:

- It should function correctly. If it does not, you may have 'translated' a piece of text that must not be translated. Locate the offending piece of text in the glossary and mark it 'Prevent Translation' (DNT).

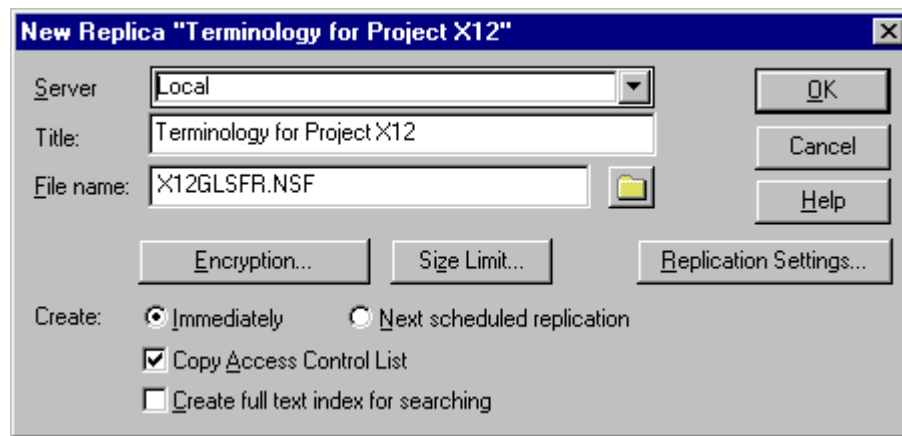
- With the exception of any words that you have deliberately marked 'Prevent Translation' (or not tagged), all text you see displayed should be in the 'pseudo' language, in other words it should be expanded or reversed. Any unmodified text that you see has not been 'translated'. Make sure that such text is tagged for translation.
- If the 'pseudo' language text is expanded text, the layout of the 'translated' output will highlight difficulties that may result from expansion in real language databases. Expansion can cause problems especially in tables and navigators.

Replicating a glossary

If translation is to be done in one or more remote sites, you can replicate the glossary database to those sites. And, by using a replication formula, you can replicate only the translation documents that are relevant to that site, along with the reference translation documents.

For example, to create a replica for French terminology you would do the following:

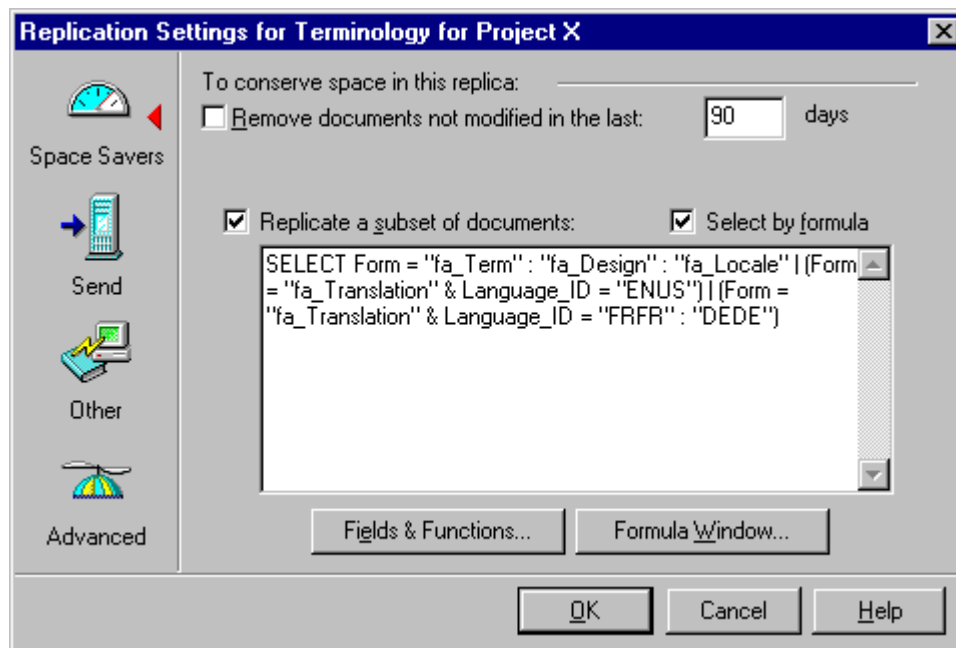
1. Install the glossary database on a Notes server.
2. Select the icon of the glossary database on your desktop.
3. Choose File - Replication - New Replica. Enter a file name for the new replica:



4. Click Replication Settings.

On the following dialog check 'Replicate a subset of documents' and enter the replication selection formula as shown. The Language_Id and Locale_Name are as specified in the table in "Adding new languages", above.

Note The various arguments are case-sensitive and should be entered using upper- and lowercase, as shown.

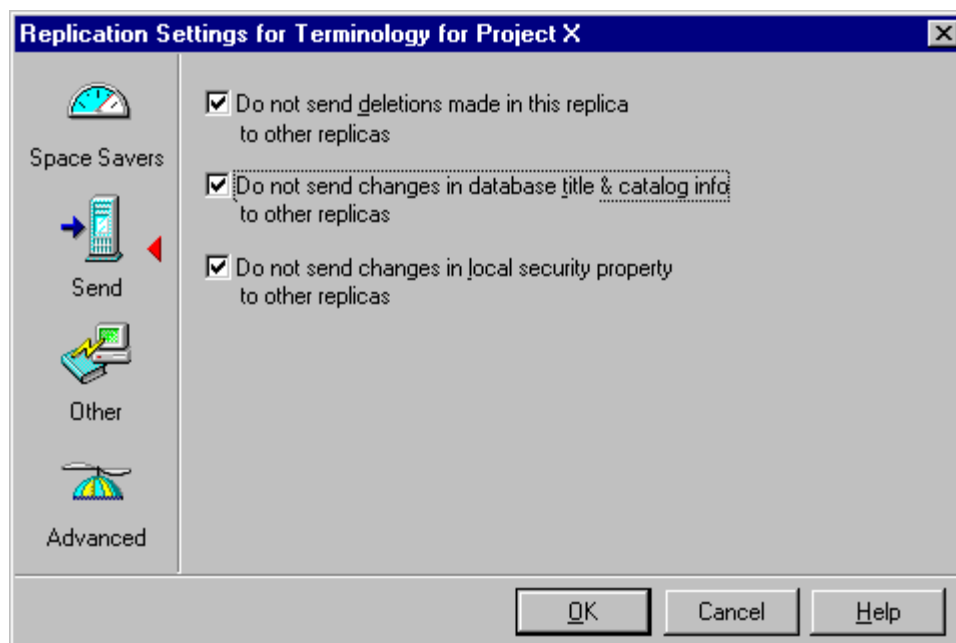


The formula shown above selects:

- All main documents (form fa_Term)
- The glossary and language information documents (forms fa_Design and fa_Locale)
- All translation documents (form fa_Translation) for the reference language, in this case US English, ENUS. Change (but do not remove) this if your source language is not US English.
- All French (France) translation documents, FRFR
- All German (Germany) translation document, DEDE

You should modify these last two entries according to the languages you want to replicate.

5. Click Send. Check all three of the boxes:



6. Click OK to go back to the New Replica dialog
7. Click OK to create the replica. The new replica is displayed on the Replicator workspace page, where you can begin replication of its contents now or later.

Once created, the new replica can be transferred to the remote site by any convenient means, for example, you could zip the .NSF file and attach it to a mail or copy it on to disks.

Caution Do not copy the database using the File - Database - New Copy command, because this produces a copy with a different replica ID. So the database will not replicate with its parent.

Establish replication in the usual way, and new terms will flow to the remote replica, and translations will flow back to the main glossary database.

Chapter 5 Translating

With Domino Global WorkBench, a Notes database or Java source is translated by translating the entries in the glossary (or glossaries) it is linked to. This is done using Notes. Once the glossary is translated you can build language databases, the databases users will work with, and localized Java resource files.

For more general information about glossary databases see Chapter 4.

Translation preliminaries

Before starting translation:

1. Comment the glossary. Good commenting can have a very significant impact on the speed and quality of translation.
2. Mark terms that must not be translated as 'do not translate'.
3. If the source database has changed since you started work:
 - If the changes are less than minor, create a new glossary and a new tagged database. Use the export and import functions to move comments and DNT information from the original glossary into the new one.

Why do this? Because it is common during a localization process to tag the source database again and again, partly for testing purposes and partly because the source database changes as development proceeds. Every time you tag or re-tag, new terms may be added to the glossary. But they are not added to the glossary in their 'natural' sequence, and it is important for translators to be able to access terms in this way (for example to see each piece of text in a form appearing together and in order). Furthermore, the tagging process cannot remove obsolete text from glossaries.

When you populate a new glossary from scratch every piece of text is added in its natural sequence. Translators can access this natural sequence when they translate by clicking 'Tagged Order Sorting' on the Translate/Review dialog. And you can be certain that only text that is actually present in the source database will end up in the glossary.

If you wish you can create a view in the glossary database that shows glossary entries in their natural order. See "To create a view with alphabetic sorting switched off", below.

- If the changes are minor, you can simply update the tagged database, although you will then find that a few pieces of text in the glossary are out of their natural sequence, as explained in the previous paragraphs.
4. Make active the languages you are translating into. See "Making a language active" in Chapter 4.

Tip It's a good idea to delay making languages active until you are ready to start translation because navigation in glossaries is faster with fewer documents.

To translate you will need:

- Either:

The glossary database. This is where you normally enter translations. If translations are to be done in one or more remote sites, you can replicate the glossary database to those sites. And, by using a replication formula, you can replicate only those translation documents that are relevant to that site, along with the reference translation documents.

or

A structured text file exported from the glossary database. After translation, the structured text file is imported back to the glossary.

See "Replicating a glossary", "Exporting" and "Importing" in Chapter 4.

- The source database, so that you can see the full context of the original text when you need to.
- (Optional) It is often helpful to run regular, intermediate builds to see how translations will look in the output. Adjusting the length of translations, where this is possible, can often make the localized output look more like the original, and limit the amount of final layout work that is needed. If you plan to do this you will need the Domino Global WorkBench software and the source and tagged databases.

Commenting glossary terms for translators

Translators involved in the localization of Notes templates and applications are not necessarily conversant with the different programming languages used in Notes (formulae, LotusScript, JavaScript, Java). Therefore, it is a good idea to help them by providing useful comments.

Developers should comment their code as much as possible, identifying what text needs to be left in the source language for functional reasons (these must be marked Do Not Translate), identifying variables in complex sentence constructs, and so on.

Even though Domino Global WorkBench provides quite a lot of context information, translators must not be expected to understand everything. Therefore, before sending glossaries for translation it is sound practice to provide some additional comments to help the translators with their task. A problem solved by a suitable comment is a problem solved for *all* the languages you are translating into.

Comments for translators should be provided by someone with a mixture of technical and translation background. This person should put him/herself in the translator's shoes and try to identify pitfalls.

How to comment?

Comments are stored in the Term description field of the Term documents. This is a multi-value field, and may contain a number of different elements.

In order to speed up the commenting process, you may like to create agents to let you add comments at the view level, in other words without opening the documents. The agents found most useful in Lotus are:

Batch comment, to apply a comment to all selected documents.

Clear, to remove comments from all selected documents.

Set comment variable, to prompt for a comment and store it in a NOTES.INI variable.

Copy, to copy the comment of the selected document and store it in a NOTES.INI variable.

Paste, to retrieve the contents of the NOTES.INI variable and apply it as a comment to all selected documents.

What to comment?

The following examples are drawn from Lotus experience in template localization:

Flag as 'translate same as', or TSA, any term which must match 100% with another term. This constraint usually arises from functional requirements. For instance, if a form is not aliased (Form Name), and there is an @command[Compose];"Form Name"), these two terms need to match both in the form properties and in the compose sentence. Or if a string refers to an element in a different database (for example the name of a view in a DBLOOKUP function), identify that reference as a 'translate same as'.

Flagging what a variable stands for. In cases like: "The required " + docname + " is not available." you should indicate what the variable/field name stands for, so that translators know how to agree the gender and number of "required", "is" and "available".

Full sentence. In a construct like "The required " + docname + " is not available.", the opening and closing text fragments are stored as separate glossary items. It will help the translators considerably if you provide the full sentence as a comment, with any variables that might be included.

When you mark a term as DNT, it is good practice to say why you are doing so in order to keep a record.

Warn translators about length limitations, particularly when text is constricted in columns, as in tables.

Where a sentence is truncated because it is followed by a field name, warn translators what type of text follows (database title, file name or user name for example).

If you have single words which are adjectives, participles or articles (for example, keywords such as Enabled/Disabled) you should identify what word(s) they refer to, in order for translators to agree on gender, number and case.

When strings belong to a dialog box title or body, identify them as such: terms belonging to a title may be translated in a different style depending on the language.

If the string contains HTML code mixed with translatable text, flag the text which should be translated.

In case of a dynamic keyword generated by a formula, warn if part of the text is an alias that should not be translated.

If there is a particular string which appears many times, add a comment warning translators, in case they want to translate it in batch.

Terms that must not (or need not) be translated

The following types of text should not be translated:

- Text that is never displayed on the screen and has no role in the functioning of the database, for example comments from the developer of the database. Translating this text is generally unnecessary, but it does not have any adverse effect on your language (output) databases.

- Text that is never displayed on the screen but does have a role in the functioning of the database. Translating these will prevent the language databases from functioning properly, so you must locate any such text in the glossary and mark it 'Prevent Translation' (DNT, or 'do not translate').
- Text that is displayed, but is the same in all languages, for example a company or trade name.

A good way to handle such text is:

- If you can make the WorkBench ignore it automatically, do that. Then it will never be tagged and never appear in glossaries. To do this, enter terms that are to be ignored in the exclusion list on the Tagging panel of the Tagging Options dialog box. This approach can be used for aliases if you have adopted a system of standard prefixes. Include the prefixes (as a wildcard, for example fa_*) in the WorkBench's exclusion list so that references to aliases are not tagged.

Note The exclusion applies to whole pieces of text, not to individual words within a piece of text. For example, a piece of static text "Notes is a product of Lotus Development Corporation" *will* be tagged if the exclusion list contains "Notes" and/or "Lotus". It will *not* be tagged if the exclusion list contains the whole phrase or "Notes*".

For larger units of text you can use the DO_NOT_TAG paragraph style in the source database.

See "Using aliases" and "Excluding text from translation" in Chapter 3.

- If you cannot make the WorkBench ignore it automatically, tag it, but mark it as Prevent Translation (DNT) in the glossary. This ensures it will always be recognized, and always treated as text to exclude from translation. It *could* be left untagged, but then, whenever you created a tagged database from a source database that included the text, it would not be found, and would be treated as 'new' text again.

Caution A glossary entry can be used more than once, and the text it represents may be hidden in one place but visible to users in another. Therefore, before you mark any glossary entry DNT, make sure that there are no displayed occurrences of it in the source database. To help ensure that every piece of text gets its own individual glossary entry, check all the Context Matching boxes on the Prompting panel of the Tagging Options dialog when you populate the glossary. This may still lead to multiple use of glossary entries, because, for example, the same text may appear more than once on a form.

Because context matching does not operate when the WorkBench looks up terms in global glossaries, you can use global glossaries to ensure that the same translation is used everywhere. Similarly, marking a term DNT in a global glossary ensures that it will remain untranslated wherever it occurs (equivalent to adding the term to the exclusion list).

Graphics that do not contain text, such as backgrounds and icons, do not usually need to be translated. By default, these are tagged by the WorkBench and put into your glossary. You can mark these as Prevent Translation in the glossary using the same procedure as for a text entry.

In general the following design items should *not* be translated:

- Aliases. The WorkBench does not tag aliases for translation.
- References to aliases in formulas, LotusScript or JavaScript. By default, the WorkBench *does* tag these, so you must find them and mark them 'Prevent Translation' in the glossary. If you have adopted a system of alias prefixes (see above) finding them will be easy. Alternatively, you can make the WorkBench skip them automatically during the tagging process by using the exclusion list (see above).
- References to untranslated design element names (rather than aliases) in formulas, LotusScript or JavaScript. Some design element names (for example hidden ones) may be left untranslated. In that case references to them in formulas (if these are to the names rather than the aliases) must also be left untranslated. The WorkBench *does* tag these, so you must find them and mark them 'Prevent Translation' in the glossary.

Note If the source database uses aliases throughout its design, you will not see any references like this.

- The names of @functions, and some @function parameters such as 'NoCache'. The WorkBench does not tag @function names, and parameters that must not be translated are listed in the default exclusion list.
- Field names. In general the WorkBench does not tag field names or references to them. However, you may see field names in glossaries because some formulas require them to be entered between double quotes. If they have been given a distinguishing prefix, such as fl_, you will be able to identify them easily in the glossary and mark them DNT.
- Hidden views, hidden form names, hidden subform names, hidden agent names, hidden folder names, hidden text in forms, hidden fields in forms which have no display implications.

Note Before excluding a hidden view from translation check that it really is hidden, and not used for browser access.

- Field default values, Input Translation or Input Validation in the cases when they refer to keyword fields whose content has been properly aliased. Again, if you have adopted a system of alias prefixes (see above) you can make the WorkBench skip these automatically during the tagging process.
- Programming keywords or tags in LotusScript, JavaScript, Java or HTML. The WorkBench does not tag these for translation, so you should not see them in glossaries. However, it *does* tag any text it finds within standard delimiters, such as double quotes, and some of this may need to remain untranslated, for example parameter values, command names, variable names and references to aliases.
- Comments, for example REMs in formulas or script. The WorkBench does not tag REMs.

Caution Comment text for any agent that appears on the Actions menu is used as a long prompt and can be seen by users. It should therefore be translated.

- Inserted subform or shared field text within a host form. See "Subforms" in Chapter 6.
- Special form names beginning '\$\$', for example \$\$ViewTemplateDefault, \$\$ViewTemplate and \$\$NavigatorTemplate.
- Special view names beginning '\$', for example (\$All), (\$Profiles), (\$Calendar), (\$Drafts), (\$FolderInfo), (\$FolderRefInfo), (\$Meetings), (\$), (\$POP3), (\$POP3UIDL), (\$Sent), (\$ToDo), and all (\$VIM***) views.

- Access control lists (ACLs) and role names.
- Database names.
- Directory and folder names.
- Domino server commands such as 'drop', 'tell' or 'quit'.

For more information about aliases, about the way the WorkBench handles them, and about what should and should not be translated, see "Using aliases" in Chapter 3, and "Tagging" in Chapter 6.

Text that should be translated

The following table lists many of the items that should be translated, or which may contain text that needs translation.

Design element	Translate
Actions	Action names and formulas.
Agents	Agent names and formulas.
Folders	Folder names, column titles and column formulas.
Forms	Form names, window title formula, static text, popups, buttons, control text, section titles, help descriptions, script. Default value formulas, input translation formulas, input validation formulas, hide when formulas (ideally these formulas should refer to hidden/control, aliased keyword fields and therefore not need translation). Keywords (all keyword names should have aliases and only the names should be translated).
Formulas	Any text within quotes that does not affect functionality. Formulas will be found in many places, for example associated with actions, agents, columns, fields and so on.
Help Using and About	All Text. At the beginning of a project, it is often a good idea to mark these as 'do not translate', because they are frequently edited and modified right at the end of the project.
Navigators	Initial view or folder name. Actions: open another navigator, alias a folder, open a view.
Script	Any text within quotes that does not affect functionality.
Subforms	Translate subforms themselves in the same way as forms. Do <i>not</i> translate subforms when they are embedded in forms.
URLs	Text that is displayed by the browser.
Views	View names, form formulas, selection formulas, column titles, column formulas.

Locating text to exclude from translation

To locate terms that should not be translated, review the glossary. Note that:

- A design synopsis of the source database will list its design elements and the text they contain. See "Creating a design synopsis" in Chapter 3. If you have used standard prefixes for aliases (see "Using aliases" in Chapter 3), search for the prefixes in the glossary. If you find any matches they should be marked 'Prevent Translation'.
- If you click the 'Term' column heading you will see all terms displayed alphabetically, and uncategorized.
- The sorting of terms in glossary views is alphabetic and this is suitable for many activities. However, when looking for text to exclude from translation you may prefer to see the terms in the order they were created, and you can do this by switching off sorting for the Term column. You will then see related text displayed together (provided the glossary entries were created in order and at the same time, for example by the WorkBench).

This can be especially useful when dealing with text that was attributed in the original database, but which you have chosen to handle as plain text rather than rich text. The WorkBench splits such text into pieces in order to retain attribute changes (see "Handling attributed text" in Chapter 6).

When viewed in a glossary, with default sorting, the pieces of text are listed where they fall alphabetically. For example, you might see this:

Term	Word count
To open Help as a database, choose Help - Help Topics	26
To search for a word or phrase anywhere in Help, close	29
To suit the needs of your site, you can create a replica of the	71
underlined	1
Using this database	3

The context of the word 'underlined' is not at all obvious. However, if sorting is switched off you will see:

Term	Word count
To find a topic in the Index view of Help, click the right-hand	32
To jump from one topic to another, double-click green	9
underlined	1
text (press green text in a box for a pop-up definition of the	42
To search for a word or phrase anywhere in Help, close	29

Here, with related text kept together, it is clear that the word 'underlined' falls in the middle of a sentence that has been split by the WorkBench. Presumably the word itself was underlined, and that's the attribute that the WorkBench is preserving.

To create a view with alphabetic sorting switched off

Do the following:

1. In the glossary database, choose View - Design.
2. Make a copy of the view 'All terms and translations'.
3. Open the copy of the view and:
 - Choose View properties.
 - Remove the entry in the Alias box.
 - Switch to the properties for the Term column.
 - Go to the Sorting tab and select the 'None' radio button.
 - Click on the heading of the untitled column to the left of the Term column.
 - Still on the Sorting tab, select the 'None' radio button.
4. Save the view.

To use the new view select it from the View menu.

Preventing the translation of terms in a glossary

If you do not want a term (or group of terms) to be translated:

1. Open the glossary in Notes and select Terms with Translations view.
2. Select the term definition document(s) for the term or terms.

Note You can mark graphic entries in this way as well as text entries.

- Click the Prevent Translation button. This sets the 'Allow translation' field of all a term's associated translation documents to No. It will then not be possible to edit the translation documents.

When a term is marked in this way, you see the Do Not Translate icon (🚫) next to it in the Terms with Translations view. It also appears in the Do Not Translate view. By selecting the document and clicking 'Allow translation' you can reset its 'Translation allowed' field to Yes.

See "Terms that must not (or need not) be translated", above.

Translation views

There are five views that are useful for translators. All are accessed from the glossary's main display:

- All terms and translations. This shows all the terms in the glossary (unless they have been marked as obsolete) categorized using various of the context fields: database name, design type (for example form or view), and design type name.

Glossary terms can contain graphics as well as text. Domino Global WorkBench handles graphics in the same way as text: they are held in term definition documents and translated in translation documents. The symbol 🖼️ indicates a graphic entry; the symbol 🗨️ indicates a rich text entry.

The screenshot shows the 'Glossary for X12 - All terms and translations' window in Lotus Notes. The interface is divided into several sections:

- Views:** A list of views including 'All terms and translations', 'Available languages', 'Do Not Translate terms', 'Obsolete terms', 'Unapproved translations', 'Untranslated terms', 'Glossary information', and 'Languages'.
- Actions:** A list of actions including 'Export', 'Import', 'Retrieve...', 'Pseudo Translate', 'Selected Documents', 'All Documents', 'Delete Duplicates', 'Delete Orphan Translations', and 'Delete Non-Linguistic Conflicts'.
- Term List:** A table showing terms and their counts. The table has columns for 'Term' and 'Count'.

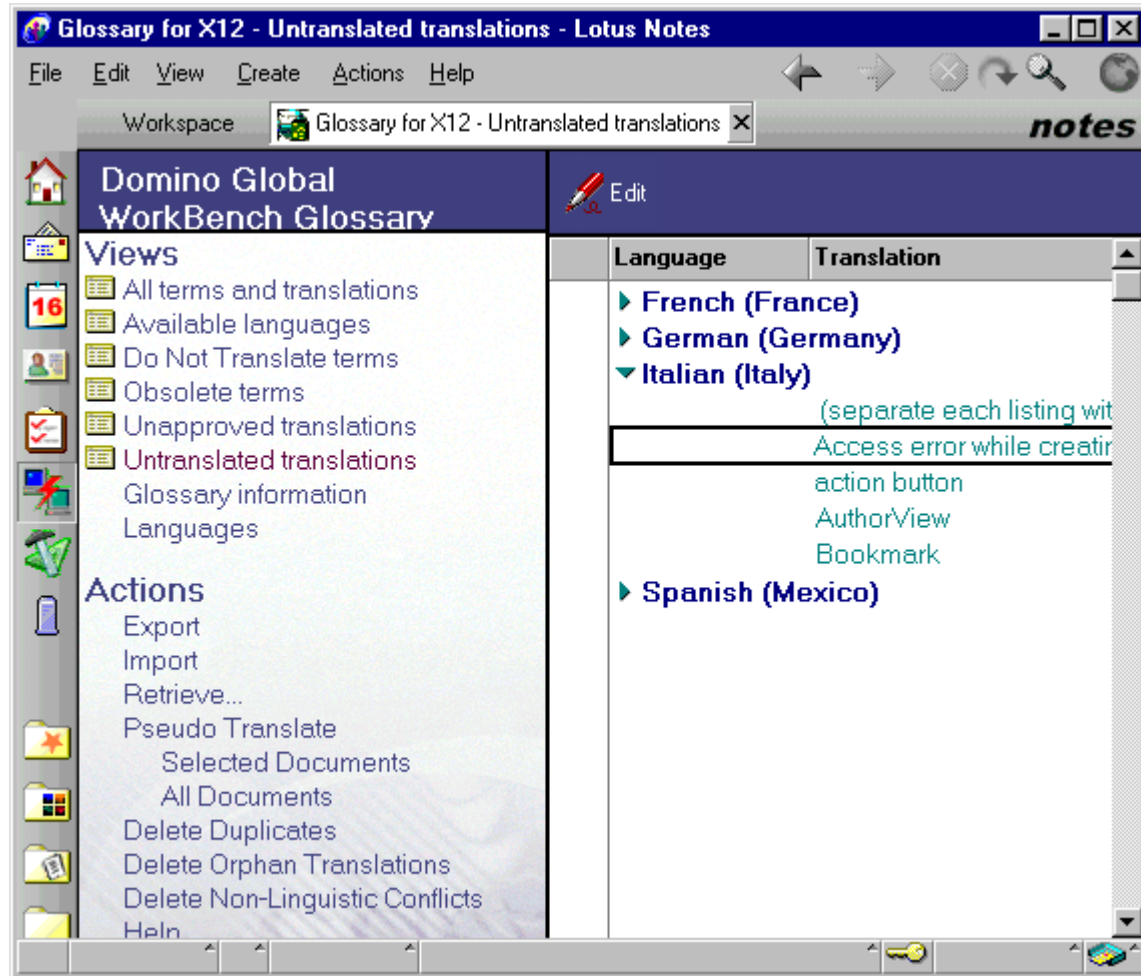
Term	Count
▼ X12 Discussion	56
▶ Agents	35
▶ Folders	36
▼ Forms	16
▼ Anonymous Response To R99	2
All Documents	2
English (United States) :	
French (France) : Tous d	
German (Germany) : All D	
Italian (Italy) : All Docu	
Spanish (Mexico) : All Dc	
🗨️ Anonymous	1
Anonymous Response To	4
AnonymousResponsetoRe	1
▼ Resource Images	11
▼ AddTopic.gif AddTopic.gif	3
🔒 action button	
🖼️ Object : 810B6C1D	3



Annotations on the left side of the screenshot explain various icons and views:

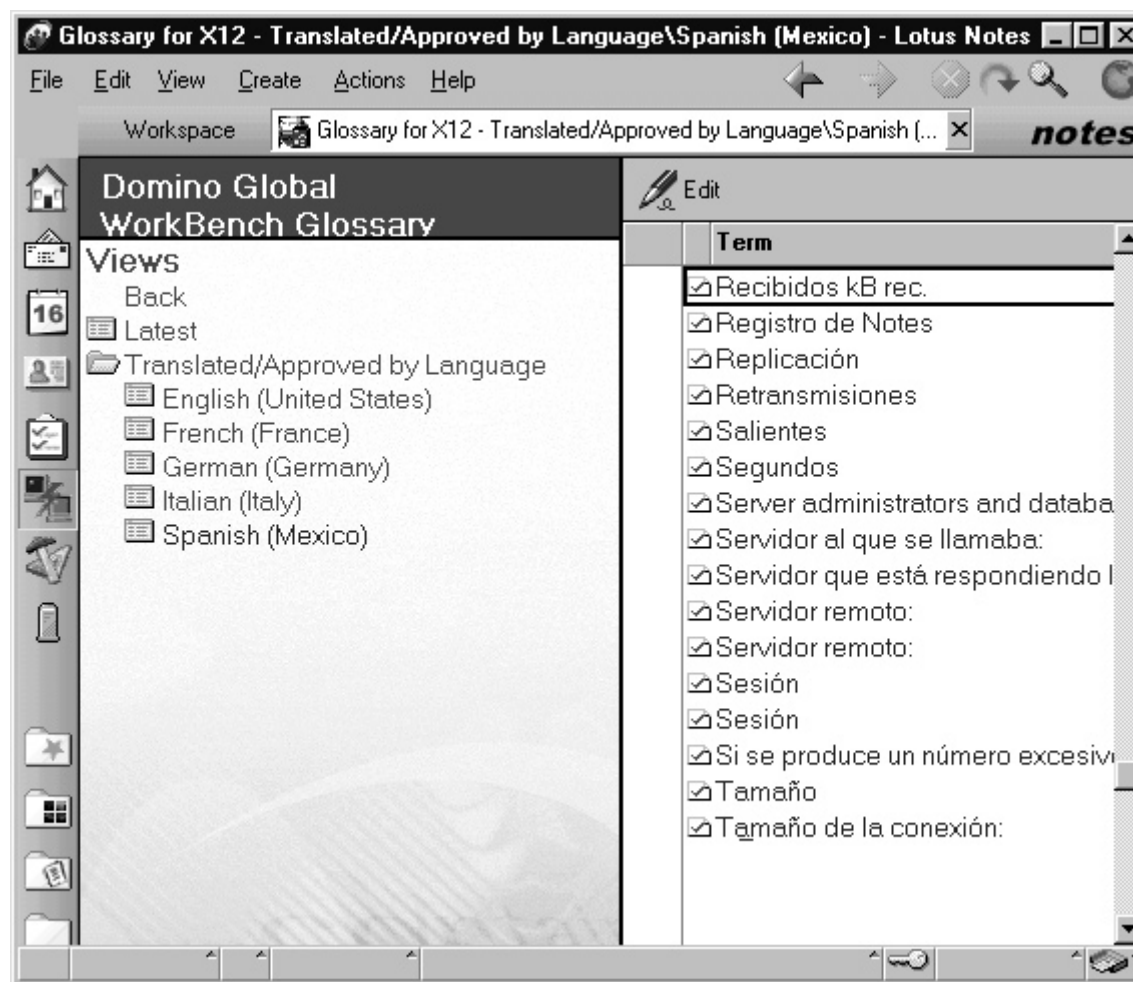
- Action buttons. To start translation, select a translation document and click Translate/Review
- Context information is used to categorize the view
- Other views
- Term definition documents
- Translation documents
- 'Translation approved' icon
- 'Translated' icon
- 'Untranslated' icon
- 'Rich text' entry icon
- 'Do not translate' icon
- Graphic icon

Note If you see any terms categorized under 'None', it may mean that the Tagger (no longer supported by Lotus) was used to create them. You can make appropriate entries in the Location fields by hand, in order that the new terms are categorized properly.

- Untranslated terms. This view shows all the terms in the glossary that have not been translated. They are listed alphabetically and categorized by language. Terms which have 'Allow translation' set to No are not displayed.



- Unapproved translations. This view shows all the translation documents in the glossary that have been translated but not approved. They are listed alphabetically and categorized by language. Terms which have 'Allow translation' set to No are not displayed.
- Do Not Translate terms. This view shows all the terms in the glossary that have 'Translation allowed' set to No.
- Languages. This displays an alphabetical list of all the translated terms in a language you select. Untranslated terms are not displayed but terms which have 'Allow translation' set to No are displayed. The usual icons indicate the status of each term: translated and approved, , or translated, .



Notes This view provides a quick way to print an alphabetical list of translated terms. Only those users who are assigned to the Approver role will see the 'approved' icon.

Caution Do not delete any of these views. If you want to modify a view, make your changes on a copy of the view.

How to translate

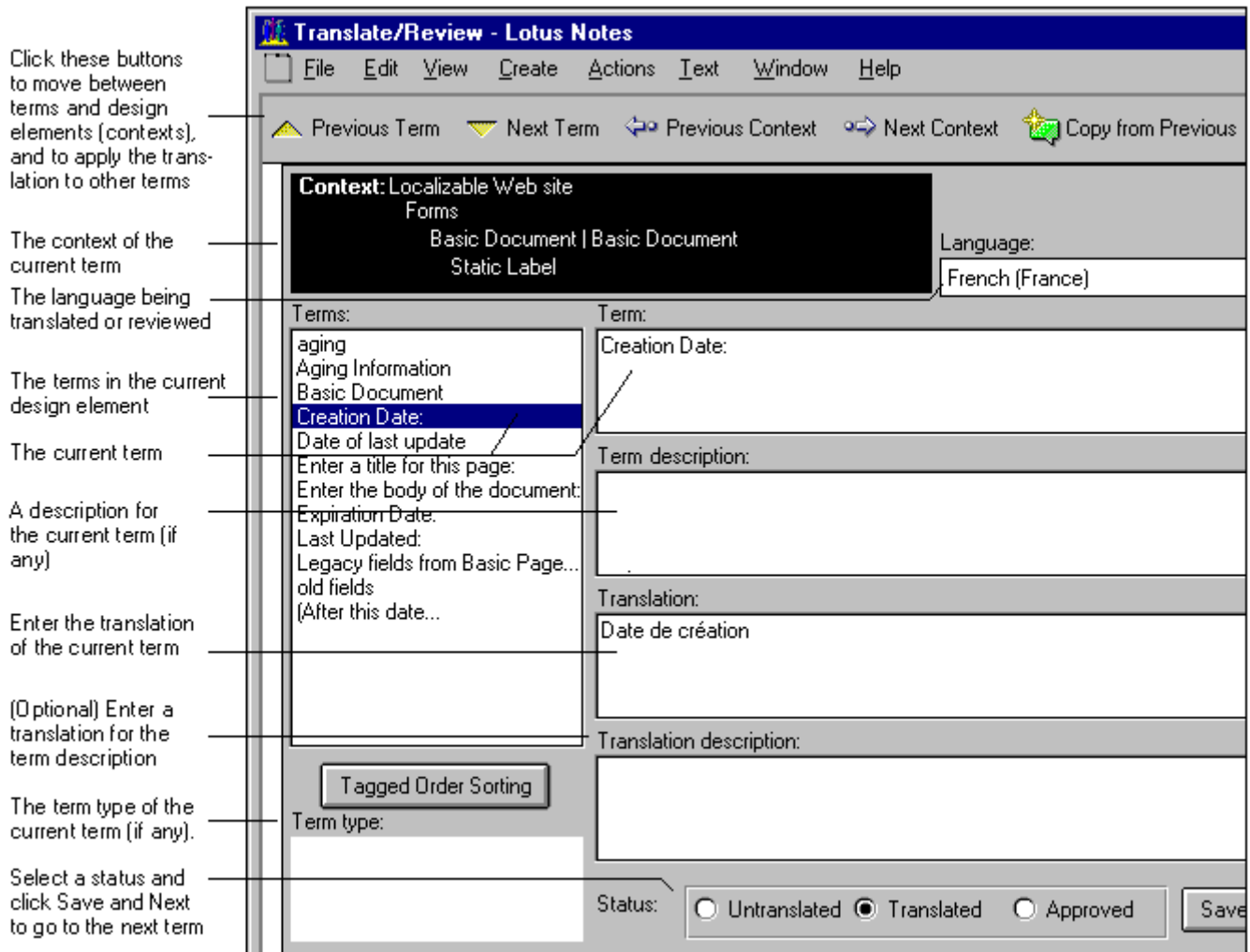
You translate glossary entries either by using the Review Translations form or by translating the appropriate translation document directly in one of the views listed above.

Using the Translate/Review dialog

To translate or review terms in a glossary:

1. In the Terms with Translations view, select the translation document you want to translate or review.
2. Click the Translate/Review button to open the Translate/Review dialog where you can quickly move from term to term in a language you choose to translate, review or edit.

Note You cannot translate translation documents that contain rich text or graphics using the Translate/Review dialog. These you must translate directly in the translation document (see the next section).



3. Select the language you want to work with in the Language drop-down box.
4. The box on the left lists all the pieces of text in the current design element. Select the one you want to translate or review by clicking on it.

Note The maximum number of pieces of text that the dialog can display is 100; use the arrow buttons to navigate to the next/previous set of 100 items.

You can click Tagged Order Sorting to display the terms in the order in which they were first entered in the glossary, rather than alphabetically. If they were entered at the same time (for example when the tagging was done) this will display the terms in their natural order.

You can also use the Previous Term and Next Term buttons to move from one term to the next, and Previous Context and Next Context to move from one design element (for example form or view) to the next.




In the 'Term Description' box you may see comments about the term entered by a developer or translation specialist before translation started. If WorkBench created the glossary entry from a piece of text in a formula, you will also see the whole formula here.

5. Translate or edit the term in the Translate box.

If you wish you can copy text from adjacent terms by clicking Copy from Next or Copy from Previous.

In the 'Translation Description' box you can add a comment for the translated term if you want to.

You can click Apply to Same to automatically apply the current translation to other terms that have the same original text (matching is case and accent sensitive). The translation is applied only to translation documents whose language exactly matches the current one. For example, if the language of the current translation document is ESES, Spanish (Spain), the translation will be applied only to ESES documents (provided the original text matches). Context is ignored.

6. Select the radio button for the status you want to assign to the term: Untranslated () , Translated () , or Approved () .

The Approved status allows reviewers to keep track of the terms they have reviewed and approved. Users with the rTranslator role cannot set a translation to Approved.

7. Click Save and Next to save your changes and go to the next term.
8. Click Done when you have finished translating or reviewing.



Translating a translation document directly

You can translate any translation document directly in Notes, and you must use this method for:

- Graphic (bitmap or native image) documents.
- Rich text documents.

The steps are:

1. In the All Terms and Translations view, open the translation document of the term you want to translate or edit, and then click Edit.

Note If you see the symbol  next to a term, you will not be able to translate it. This indicates that the term's 'Allow translation' field has been set to No and that it must not be translated. Also, you cannot edit the translation document for the reference language. The symbol  indicates a bitmap entry.

Translation

Details | Translation Settings | Advanced Options

Language: French (France)
 Original term (Plain text): All documents
 Term description:
 Translated term: 『Tous documents』

Details | Translation Settings | Advanced Options

Translation allowed: Yes
 Translation description: 『』
 Term type(s):
 Translation status: Translated

Details | Translation Settings | Advanced Options

Apply to Language Variants

Apply to Same

2. Enter your translation in the 'Translated term' field.
 If the field contains a graphic, follow the guidelines below.
 If the field contains rich text, you should apply the attributes (for example, bolding, underlining or font changes) appropriately to the text you enter or edit.
Note If the parent term document was created by the WorkBench during tagging, and the term is part of a formula or a piece of script, the 'Term description' field, above, will show the term in context.
3. (Optional) Enter a comment for the translation in the 'Translation description' field on the Translation Settings tab.
4. (Optional) Click Mark Translated to set the status of the translation document to 'Translated'.
5. (Optional) Click one of the Apply Translation buttons on the Advanced Options tab if you want to apply the current translation to other documents.
 'Apply to Language Variants' applies the translation to other translation documents for this term, provided their basic language is the same as the current one. For example, if the language of the current translation document is ESES, Spanish (Spain), the translation will be applied to ESAR, Spanish (Argentina), documents and ESMX, Spanish (Mexico), documents.
 'Apply to Same' automatically applies the current translation to other terms that have the same original text (matching is case and accent sensitive). The translation is applied only to translation documents whose language exactly matches the current one. For example, if the language of the current translation document is ESES, Spanish (Spain), the translation will be applied only to ESES documents (provided the original text matches). Context is ignored.
6. Click Close.

Translating graphics

Notes Release 5 supports .GIF and .JPEG native images as well as bitmap (.BMP) images. Lotus recommends that you translate graphics at the very end of your localization process, as there is no quick way of updating them.

To translate a graphic:

1. Create a translated version of the image. If you can, do this by editing a copy of the graphic file that was used in the source database design.

Note If necessary, you can copy the original graphic from the Notes document and paste it into a graphics editor, but be aware that if you do this you may lose some graphical information, as the clipboard uses the .BMP format.

2. Save the translated image to a file of the appropriate format.

The WorkBench will build images to language databases using whatever image format it finds in the glossary. So, if you want to ensure that the format of built images is the same as that of the source image, you must import the translated image to the glossary in the appropriate format. In other words, if the original image was a JPEG image you should import its translation to the glossary from a .JPEG file.

Caution In navigators, the only graphic format that is supported is .BMP. Therefore, it is especially important to translate these images as .BMP, and *not* as .GIF or .JPEG.

3. In Notes, open the glossary database and then the translation document for the graphic.
4. Delete the original version of the graphic from the translation document and import the translated version (File - Import). You should always use the File - Import command (not paste) to bring in the localized version.
5. Save the translation document.

Duplicate glossary entries

You may find several glossary entries for the same piece of text. This is because the text appears in several different places in the source, and the Workbench has taken account of context, so a term from a form (for example) is regarded as being different from the same term in a view, even though the text itself is identical.

It is important to maintain this distinction because in some languages the translation for a term may vary from context to context.

If you have many of these entries, you may want to create an agent to copy a translation from one translation document to the next.

Chapter 6 Using the WorkBench

Most Domino Global WorkBench processes are carried out using the WorkBench itself. However, before you use the WorkBench to localize a Notes database you should:

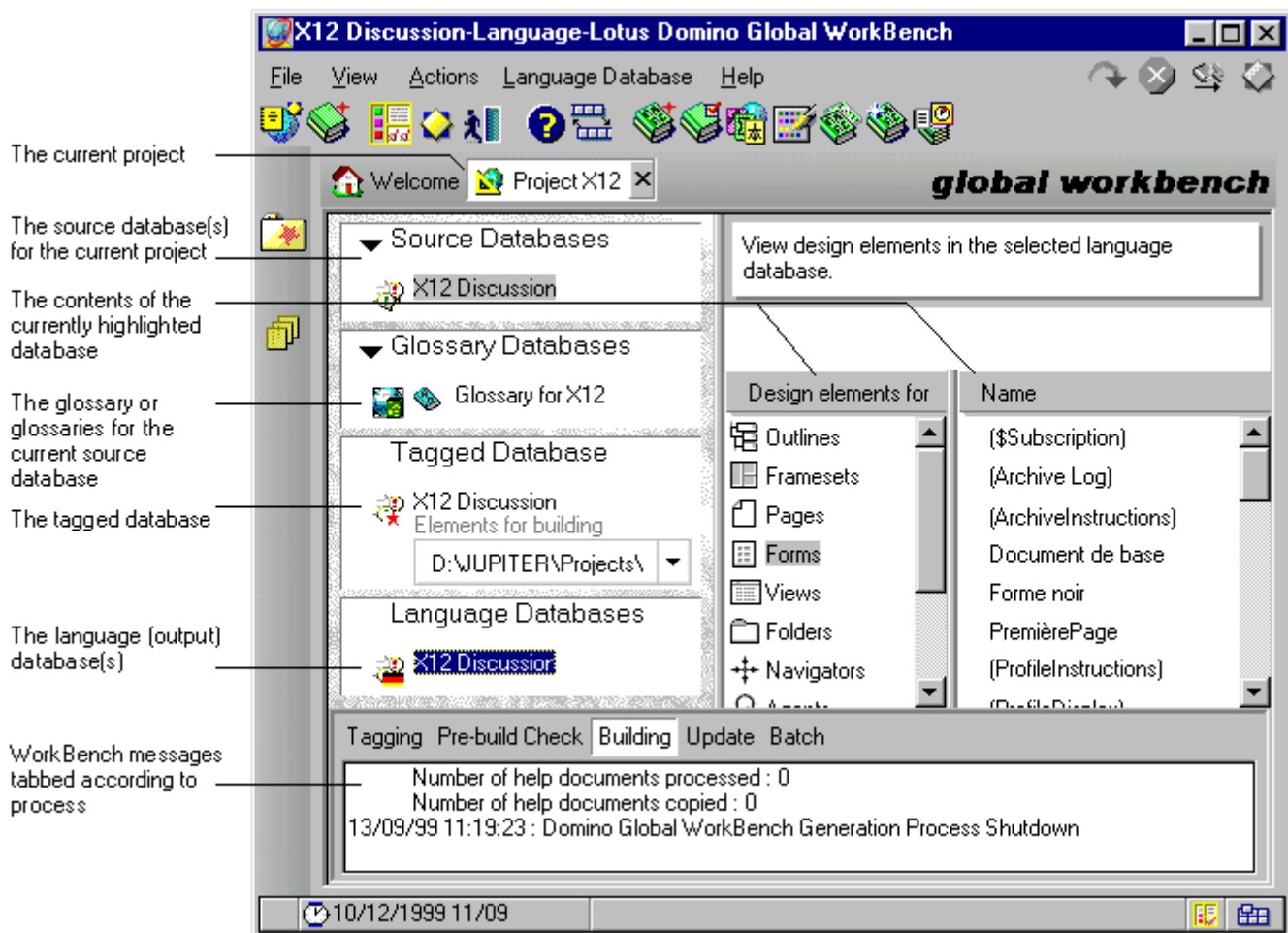
- Prepare the source database (see Chapter 3).
- Prepare glossaries (see Chapter 4).

For a description of the overall process, see "The overall Domino Global WorkBench process" in Chapter 1.

For a description of how to use the WorkBench to add language synchronization features to a database, see "Adding language synchronization features" in Chapter 8.

The WorkBench display

When you start the WorkBench you will see the main display. This illustration shows the display with a project loaded. To open an existing project you click a project icon on the left edge of the display, and then click on a source database to load its associated glossary, tagged and language databases.



Tips

When running the WorkBench you can right-click an item or a pane to see a pop-up menu of relevant options. And you can launch any database into Notes by double-clicking it. To see information about a particular design element (for example its fields) right click it in the Name pane to open its properties box.

Unicode

The WorkBench can display text using Unicode fonts provided they are installed on your computer. Notes Release 5 includes Unicode Monotype WorldType fonts as an option in Install. See "Unicode" in Chapter 1.

Projects

A Domino Global WorkBench 'project' is a group of one or more database application(s) that you want to localize. These are called 'source databases'. A project can consist of one source database or several. Each source database within a project can have a number of other databases associated with it:

- A tagged database. This provides the essential links between the pieces of text in the source database and entries in glossaries.
- One or more glossary databases. These hold the text that the WorkBench uses for building. Translation work is done in glossaries, not in any of the other databases.
- One or more language databases. These are the output databases that people will use. They are built from the tagged database and associated glossaries.

The WorkBench stores information about projects in a project database (PROJECT.NSF) held in the Domino Global WorkBench program directory.

Creating a project

To create a project:

1. Start the WorkBench from the Domino Global WorkBench folder or, if the WorkBench is already running, go to the Welcome display.

Notes If you have installed Domino Global WorkBench for the first time, a prompt will tell you that the project database, PROJECT.NSF, cannot be found and ask if you want to create it. Click Yes to proceed.

If you are reinstalling, the project database will already exist. To update it, replace its design from the newly-installed template, DGWPROJ.NTF, in the Notes data directory.

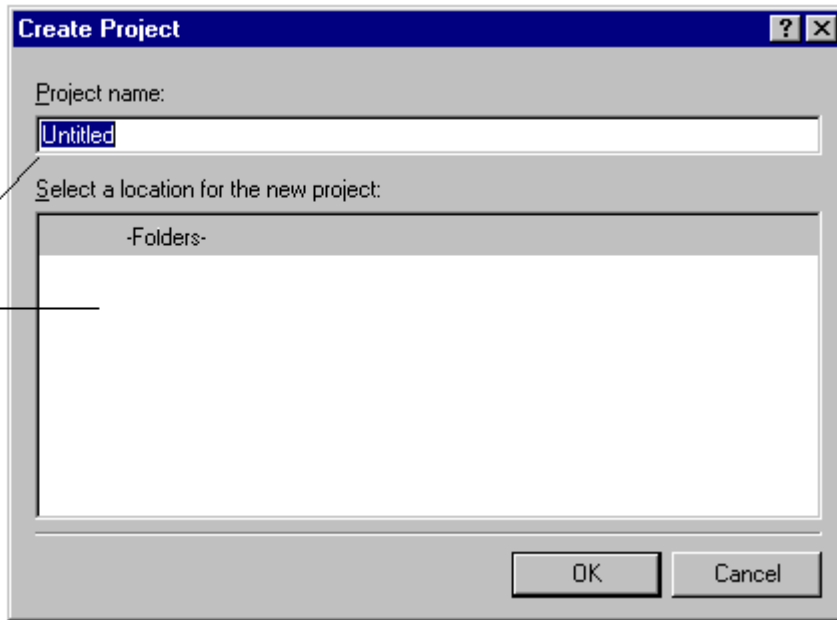
2. On the Welcome display click 'Create a New Project'. The Create Project dialog box will open.
3. Enter a name and location (folder or subfolder) for the new project.
4. Click OK.

Note You can also use the icons on the left edge of the main display to create a project. See "Opening an existing project", below.

Project details are held in the database PROJECT.NSF, in the program directory

A name for the new project

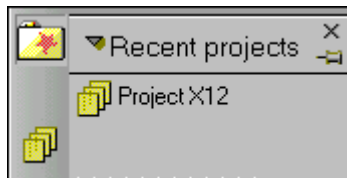
A location for the new project







Opening an existing project

To open a project:

1. Start the WorkBench from the Domino Global WorkBench folder. You will see the WorkBench main display. The folder icon at the left edge of the screen displays a list of your most recent projects:



Click the project you want to open. Bookmarked projects and folders are indicated by pages and folder icons under the main folder icon. To open a bookmarked project click the appropriate pages icon, .

To create a new project, click  and choose Create New Project (project details are stored in PROJECT.NSF in the program directory). To create a new folder, click  and choose Create New Folder. To display the folder display permanently click the pin icon or click  and choose Pin On Screen.

Right-click a project bookmark to see a menu of project options: Open, Remove, Rename and Change Icon.

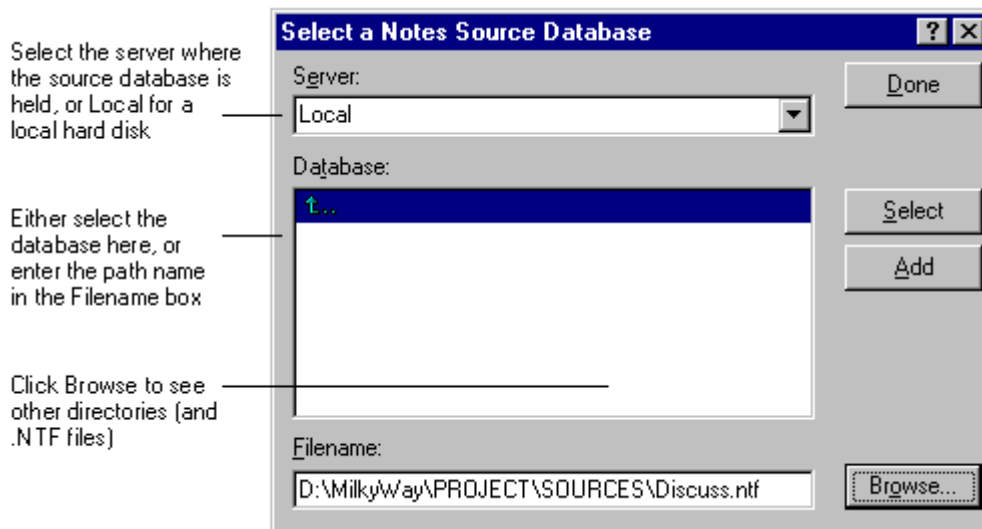
2. After opening a project, select one of its source databases to display the glossaries, tagged database and language databases associated with that source database.

Loading source databases

Before loading a source database, check that none of the design elements that you want to localize have been marked 'Prohibit design refresh or replace to modify'. See "Design elements marked 'Prohibit refresh or replace'" in Chapter 3.

To work on a new source database you must load it into the WorkBench as part of a project. The steps are:

1. Start the WorkBench from the Domino Global WorkBench folder.
2. If the project for the new source database is not already displayed:
 - Select it by clicking the folder icon at the left of the display.
 - or
 - Create a new one.
3. Choose File - Add Source Database, or click the 'Click Here' prompt text, which appears if you have created a new project.
4. Complete the Select a Notes Source Database dialog box to select a database. You will see the new source database displayed in the WorkBench main display.
5. Repeat steps 3 and 4 if you want to add further source database(s).



Loading glossaries

Before you can create a tagged database you must load at least one glossary. The steps are:

1. Start the WorkBench and highlight the source database.
2. Choose File - Add/Create Glossary Database, or click the 'Click Here' prompt text, which appears if no glossary is selected.
3. Complete the Select a Glossary Database dialog box to select one or more glossaries.

If you want to create a new glossary from scratch, click Create.

When adding a glossary you can specify it as local (🏠), project-wide (📁) or global (🌐). A local glossary is used for the current source database only; a project glossary is used for all source databases in the current project; a global glossary is used for all source databases in all projects. The WorkBench consults them in that order: local first, global last.

You can remove a glossary from a source database by highlighting it and pressing the Delete key or right-clicking and choosing Remove From Project. If you remove a global glossary, it will be removed from all projects. If you remove a project-wide glossary it will be removed from all source databases in the project. If you want to remove a non-local glossary from just one source database, you must first change its type to local. To do this, right-click and choose Properties. Then select 'Local'.

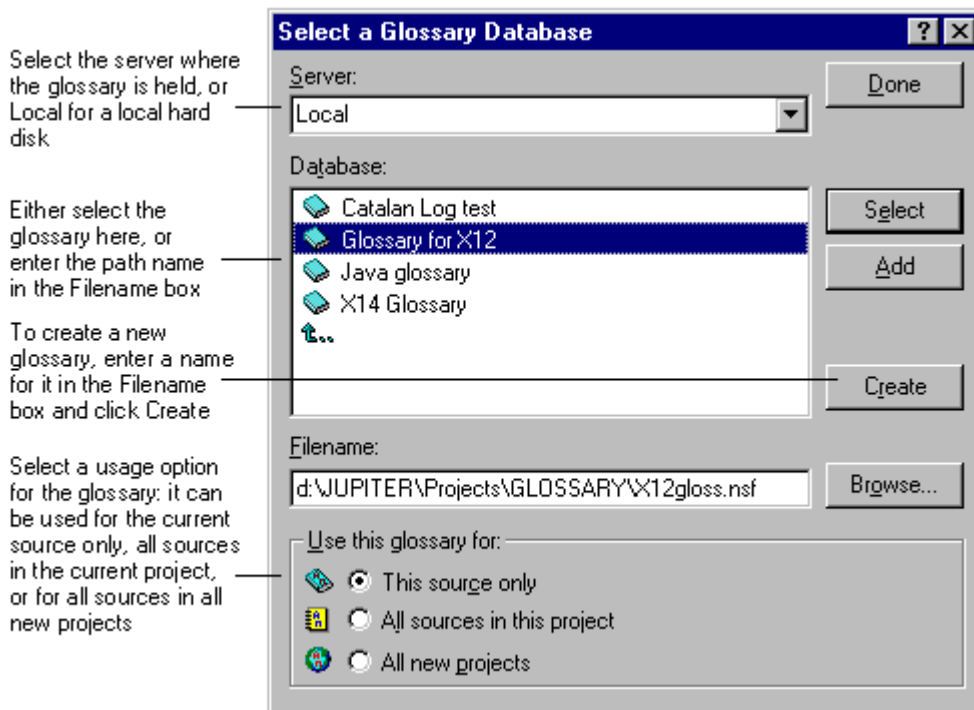
- Click Select to close the dialog box after adding a glossary. Click Add to keep the dialog box open so that you can add another glossary if you want to.

Now that at least one glossary has been assigned, you can view the source database's design elements in the WorkBench display. Simply highlight the source database and select a design element type (in the Design elements pane). You will see the design elements of that type.

If you have several glossaries of one type, they will be consulted by the WorkBench in the order in which they are displayed on the screen. To change this order, right-click a glossary and select Properties to open its infobox. Then click the up or down arrow to change the priority. You can change the order of a glossary only within its type (global, project or local).

Caution Do not use the same glossary to build Release 4.6 and Release 5 language databases, because the two releases store rich text and graphics entries differently. Instead, create a copy of the glossary you would like to use. Then, if you are using a Release 5 glossary to build 4.6 databases, put the copy on a Release 4.6 server and retranslate its rich text and graphics entries using a 4.6 client; and if you are using a Release 4.6 glossary to build 5.0 databases, put the copy on a Release 5 server and (optionally) retranslate its graphics entries using a 5.0 client to take advantage of Release 5's improved graphics formats.

See "Using Domino Global WorkBench with Notes Release 4.6 databases" in Chapter 1.



Scandinavian collation

The WorkBench cannot correctly locate text in a glossary database if the glossary uses Scandinavian collation and the text in your source database contains any of the following characters: ä, Ä, Å, å, Å, ö or Ö. To switch off Scandinavian collation:

- Start Notes on the client or server where the glossary is held.
- Choose File - Tools - User Preferences - International and uncheck 'Scandinavian collation'.

3. Restart Notes.
4. Open the glossary and select the 'Translated by language' view for the language of your source database. Press Shift-F9 to rebuild the view.

Creating a tagged database

Once you have loaded a source database and a glossary, you can create the tagged database.

Note The WorkBench's automatic tagging will be much more accurate if everything in the source database has been aliased, and a standard system of prefixes has been adopted. See "Using aliases" in Chapter 3.

The WorkBench creates the tagged database by looking up each piece of text from the source database in the loaded glossary or glossaries. You can run the procedure with varying degrees of prompting:

- If you run without any prompting and with the automatic creation of new terms in the glossary, whenever the WorkBench finds a matching glossary entry it creates a tag to it; if it does not find a match it creates a new glossary entry and tags to that.

When creating a tagged database for the first time, it is simplest to choose this fully automatic option.

- If you run with full prompting, you decide when the WorkBench should create tags and/or new glossary entries.

You need at least Author access rights in the glossary database in order for the WorkBench to create new glossary entries. And if you run the process with any degree of prompting you will need to understand about tagging. See "Tagging", below.

The steps are:

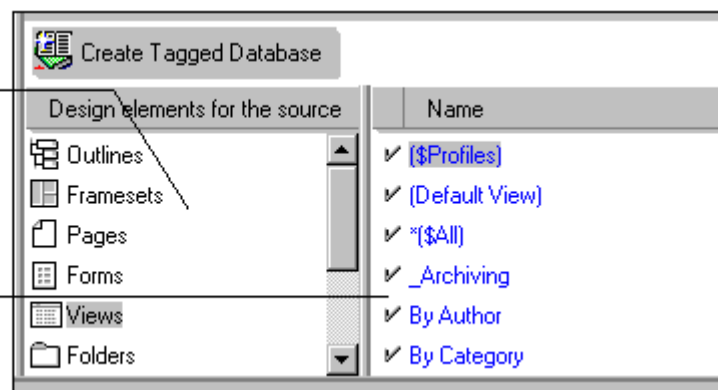
1. Highlight the source database in the WorkBench display. You will see its design elements displayed in the Design Elements and Name panes. Select the elements you want to tag by clicking them in the Name pane to add or remove tick marks. Ticked elements will be tagged. If there are elements which you know will not be localized, or which you wish to localize manually, you can exclude them from the tagging process by deselecting them here.

Right-clicking in the 'Design elements' or 'Name' pane displays a menu that allows you to select or deselect all design elements, all design elements of the current type, or open the properties box so that you can see information about a particular design element (for example its fields).

Select a design type in the pane on the left. In the pane on the right you see a list of the design elements of that type

Click design elements on the right to select or deselect them for the tagging process

Selected items have a tick



Note Help documents (data documents) are not, by default, selected for tagging.

- Click the Create Tagged Database button or the 'Click Here' prompt, or choose Actions - Create Tagged Database.

Note If the tagged database already exists you should use Update Tagged Database to bring it into line with the source database. Alternatively, you can use Re-create Tagged Database to rebuild it, but if you do that any decisions you originally made in response to prompting from the WorkBench will be lost. These options are available on the Tagged Database menu which appears when the tagged database is highlighted.

See "Updating", below.

- Complete the New Database panel of the Tagged Database Creation dialog box.

Select the server where the tagged database is to be created, or Local for a local hard disk. If your source database is a 4.6 database you must select a 4.6 server

Enter a path name for the tagged database (click the button to select a directory)

- Complete the Basics panel of the Tagged Database Creation dialog box.

Select the language of your source database

Select a glossary. New terms created automatically by WorkBench will be entered here

If you want to restrict the process to terms belonging to a particular term type, select it here

Choose the level of error and warning messages you want to see

Notes The Term Type is an optional user-defined grouping (explained in "Term Types" in Chapter 4). If you choose a term type, the WorkBench will only search among terms assigned to that type. And any new terms that the WorkBench creates in the glossary will be assigned to this term type.

The reference language of the 'New Terms' glossary *must* be the same as the language of the source database.

The report database will be used for reports from building, as well as tagging operations. To create a new reports database, click Browse to open the Select Reports Database dialog and then click Create.

5. Complete the Lookup panel of the Tagged Database Creation dialog box.

The screenshot shows the 'Glossary Lookup Options' dialog box with the following settings and annotations:

- Glossary Lookup Options:**
 - Case sensitive (Annotation: Select the matching options for the tagging operation)
 - Accent sensitive
 - Combine broken terms (Annotation: Select Combine broken terms to remove invisible attribute changes before looking up or creating terms)
 - Rich text sensitive (Annotation: Select Rich text sensitive to match attributed text against rich text glossary entries, and to create rich text glossary entries for such text)
 - Limit of terms to lookup: 999
- Context Matching:**
 - Database name
 - Type
 - Type title
 - Subtype
 - Subtype title
 - Archive
 - Property bundle file
 - Resource key
- Glossary Marking:**
 - Mark terms used during tagging (Annotation: If you check the 'Mark terms' box, the text in the 'Marker text' box will be added to the glossary entry of every term that is used during tagging. This can help you to locate obsolete glossary entries)
 - Marker text: 09:23, Thursday, January 20, 2000

Additional annotations on the left side of the dialog:

- If you check any of these, matching will be based not only on text but also on the context of the term (pointing to the Context Matching section)

Notes Check 'Case Sensitive' if you want glossary lookups to take account of case (to regard upper- and lowercase versions of a letter as different characters). You might check this option if your source database uses special keywords.

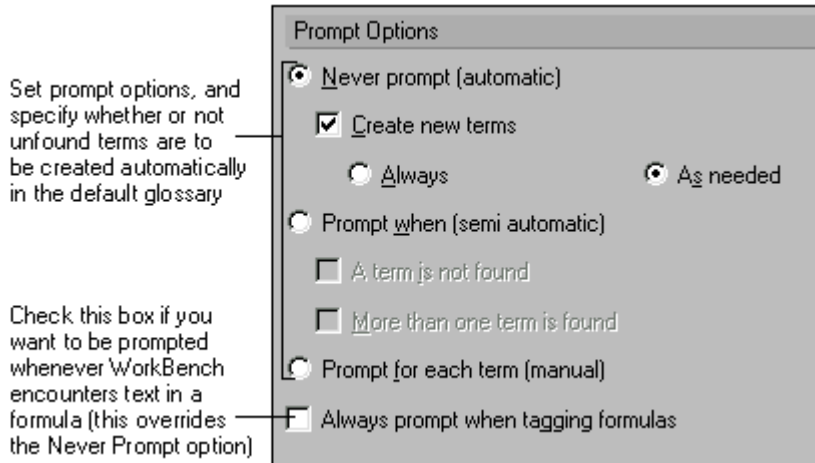
Check 'Accent Sensitive' if you want glossary lookups to take account of accents (to regard differently accented versions of a letter as different characters). For an English source database you should not need to check this option.

Check 'Combine Broken Terms' and 'Rich Text Sensitive' if you do not want the WorkBench to split sentences at font or text style changes. This will create RTF documents in your glossary, which you will have to translate individually (not using the Translate/Review dialog box). For more information about the 'Combine Broken Terms' and 'Rich Text Sensitive' options, see "Handling attributed text (rich text)", below.

Normally the WorkBench will base its searching only on text, but you can choose to have it pay attention to the context as well. See "Location (context) fields" in Chapter 4 (however, when looking up text in *global* glossaries, the WorkBench *never* pays attention to context). In the illustration shown above, all context options have been selected. This means that a separate glossary entry will be created for each occurrence of a piece of text, except where it appears more than once in the same sub-element. The reason for doing this is to give maximum control over translation; some occurrences of the same text may need to be translated differently in some languages. Checking all options except Database Name often gives good results.

When using marker text, note that during an update, elements that have not changed are not retagged and so any marker text will *not* be added to terms they use. In general, marker text is most useful when you are tagging or retagging a source database from scratch. If several source databases use the same glossary, you will have to retag each one of them in order to discover obsolete terms. Adding marker text slows down the tagging process significantly.

6. Complete the Prompting panel of the Tagged Database Creation dialog box.

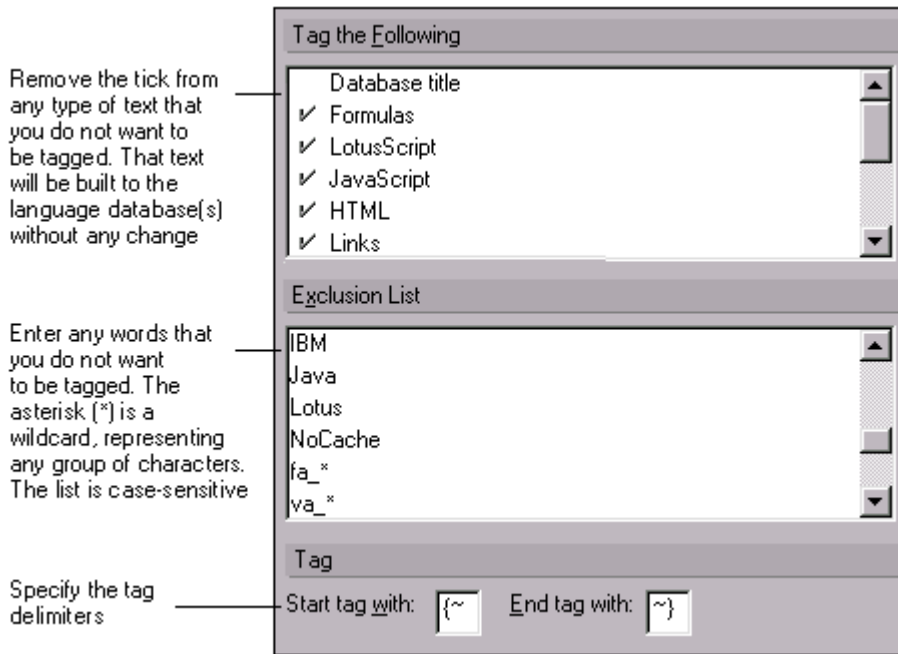


Notes When adding terms to a glossary automatically, the WorkBench assumes that the language of the source database is the reference language of the glossary, and it creates glossary entries on that basis. Therefore, you should use the 'Create new terms' option only if the language of the source database and the glossary's reference language are the same.

If you select 'Never Prompt', WorkBench will tag to the *first* piece of matching text it finds in the glossary or glossaries. If your Lookup settings lead to multiple matches the tagging may not be what you expect. For initial tagging Lotus recommends that you select 'Always', which will create new glossary entries for all the terms in the database, and will perform no lookups. For updating, select 'As Needed' which will match terms against existing glossary entries where it can, as defined in the Context Matching section.

You can use the 'Always prompt when tagging formulas' option to make the WorkBench prompt you each time it encounters text in a formula. This can be useful because text in formulas will sometimes be a displayable string which needs to be tagged, and sometimes the alias of another design element which must not be tagged. See "Formulas", below. Alternatively, you can have the WorkBench tag everything (as here) and then review the glossary entries, marking any that should remain untranslated as Prevent Translation. If you use a system of prefixes for aliases (see "Using aliases" in Chapter 3), you should not need to check this box.

7. Complete the Tagging panel of the Tagged Database Creation dialog box.



Notes You can choose not to tag certain types of text (for example, LotusScript or HTML). The entry 'Link' refers to URL and doc links, and 'Database Class Name' is relevant only when a template is being tagged: it is the name of the template. In addition to the default selections, you should usually select 'Hidden Design Elements' as these can contain visible text. See "Hidden design elements" in the "Tagging" section, below.

If you have adopted a system of prefixes for alias names (see "Using aliases" in Chapter 3), you prevent them from being tagged by entering the prefixes in the (case-sensitive) 'Exclusion List', followed by an asterisk. For example `fa_*` prevents the tagging of any text that has the prefix 'fa_'. This provides a reliable way of preventing the tagging of alias names in formulas or LotusScript. If the source database includes either of the functions `@Command([OpenNavigator];"navigator")` or `@Command([ViewChange];"view")` you may need to make additional entries to this list. See "Exclusion list entries", below.

The only wildcard character supported by the WorkBench is the asterisk (*), and it must appear at the end of the string.

See also "Exclusion list entries", below.

Under 'Tag', check that the tag delimiters are unique: they must not appear anywhere in the design of your source database. If necessary, change them. Use at least two characters for the start and end delimiter. The defaults are `{~` and `~}`. In Lotus databases, `^}` and `{^` work well as alternative opening and closing delimiters. Do not use `[` or `]` as tag delimiters because Domino uses them to mark HTML tags. Once a tagged database has been created you cannot change its tag delimiters.

The open and close brace characters, `{` and `}`, are used as literal string delimiters by LotusScript. If you use these characters in tag delimiters for the WorkBench, you may see some anomalies when you view the tagged database. However, this will not cause problems in the built language databases.

8. Click OK, and the WorkBench will begin to create the tagged database.

- If you have chosen not to be prompted about tagging, the WorkBench will tag source text to the first matching glossary entry it finds. It will create new glossary entries if that is what you specified in Step 6.

If you have chosen to be prompted about tagging, and the WorkBench finds one or more matches for the source text in the glossary or glossaries, you will see the Lookup dialog box. Here, the WorkBench presents terms according to what you specified above in Step 5. It also presents glossary terms if only *part* of their text, starting at the beginning of the term, is identical to that of the text in the source database. You choose whether or not to tag terms or create glossary entries. Highlight a term in the Found Term(s) box and click Accept Found to tag the source text to that term.

The term from the source database that is currently being looked up in the glossaries (shown in context)

Terms found in the glossaries that match the text (and context, if that was specified in tagging options) of the current term

The context of the term currently being looked up and the currently highlighted found term

Click Accept Found to use the selected term in the tagged database. Click Skip to leave the term untagged

Comments (if any) for the highlighted found term

Available translations and status for the highlighted found term

Lookup

Terms and Contexts

Source Term within context

Found Term(s)

StandardTitle := Subject + @DocDescendants[" (@If(@IsNewDoc; "New Topic"; @ViewTitle = "A

New Topic
New topic type

View Term in Larger Window...

Context	Source Context	Found Context
Notes		
Database	X12 Discussion	X12 Discussion
Type	Forms	Forms
Type Title	Main Topic fa_Main	Main Topic fa_Main
SubType	Window Title	Window Title
SubType Title	N/A	N/Ae
Java		
Archive		
Filename		
Key		

Found Term Translation Settings

Description

StandardTitle := Subject + @DocDescendants[" (No Responses]" @If(@IsNewDoc; "New Topic"; @ViewTitle = "AuthorView"; Sub

Translation Status

English (United States)
 French (France)
 German (Germany)

Notes If you choose a term whose text is different from the term in the source database, your tagged database and the language databases built from it will be different, in that respect, from your source database.

If none of the matches is suitable you can click Create New to create a new term in one of the glossaries (using the current text from the source database), and tag to that. Or you can click Skip to leave the text untagged.

If you chose 'Rich Text Sensitive' in Step 5:

- The WorkBench matches pieces of rich text from the source database against *rich text* entries in the glossaries. And only *exact matches* will be presented in the Look up panel (attributes must match as well as text). If you select one of the matches (you can preview it in the Rich Text Preview dialog box by clicking Rich Text) the text will be built to the language database(s) as rich text.
- The WorkBench matches pieces of plain text from the source database against *all* entries in the glossaries. Any entries whose *text* matches the source text will be presented in the Look up panel. If you select a rich text match (you can preview it in the Rich Text Preview dialog box by clicking Rich Text) the text will be built to the language database(s) as rich text. If you select a plain text match the text will be built to the language database(s) as plain text.

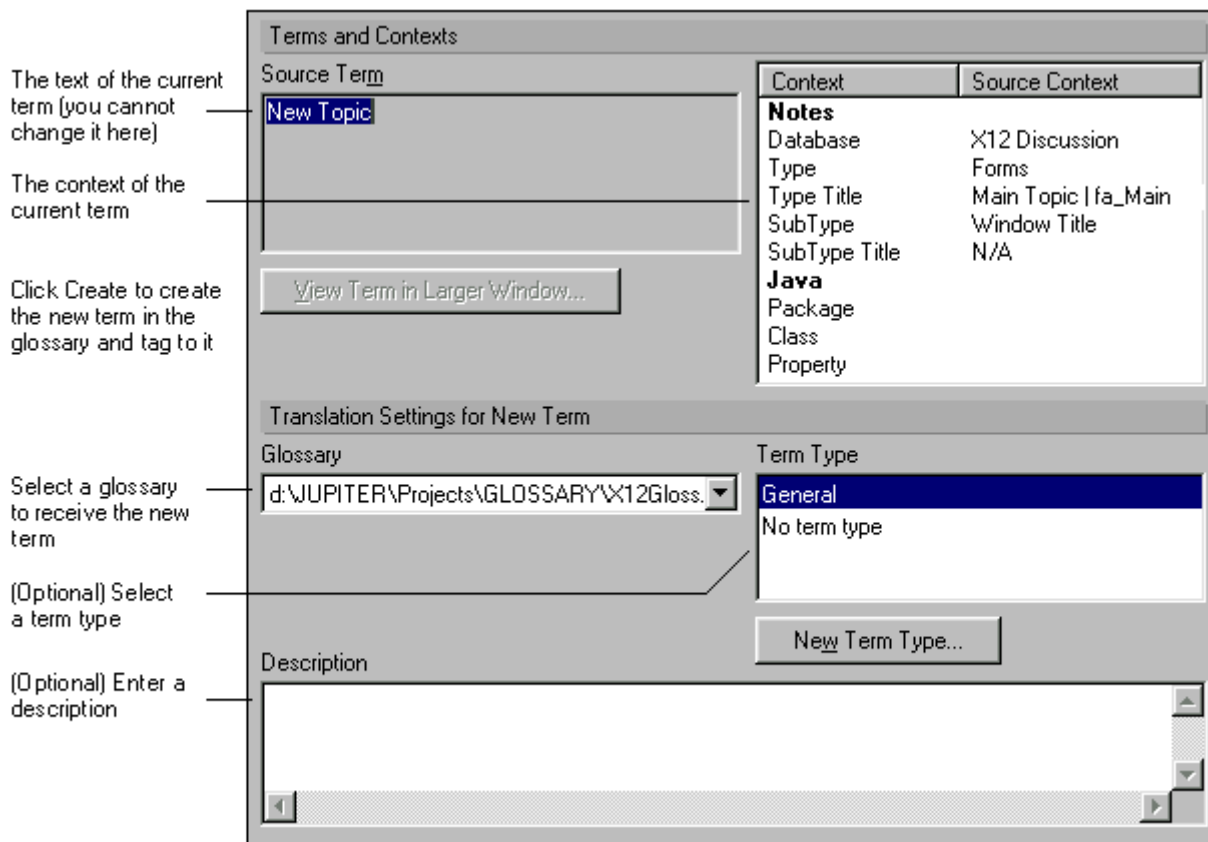
For a rich text term, the context Subtype is 'Static Rich Text'.

If you did not choose 'Rich Text Sensitive' in Step 5:

- The WorkBench will match all pieces of text from the source database against *all* entries in the glossaries. Any entries whose *text* matches the source text will be presented in the Look up panel. If you select a match the text will be built to the language database(s) as plain text.

See "Handling attributed text (rich text)", below.

If you have chosen to be prompted about tagging, and the WorkBench does not find a match for the source text in the glossary or glossaries, you will see the Create New Term dialog box.



Notes To create a glossary term and tag to it, click Create.

The Create New Term dialog does not allow you to create a term whose text is different from the source database text.

The reference language of the glossary *must* be the same as the language of the source database.

Terms created in this way will be created as plain text terms. See "Handling attributed text", below.

Exclusion list entries

In the Exclusion List, you should add terms which appear frequently in your source database, but which do not need to be translated. Customize the list according to your requirements. The more terms you have in here, the fewer terms you will have to mark as DNT (do not translate) in your glossary. Remember that you can use an asterisk as a wildcard at the end of an exclusion list entry. If your application uses conventions for prefixing design elements you could define `fa_*` to bypass all form names and reference to them which begin with the `fa_` prefix, for instance.

A sound piece of advice is to tag one database, if you are working on a full application, and analyze the DNTs you have; if you identify a clear pattern, or a particular term which appears over and over again and which is systematically DNTed, then these would be good candidates for the exclusion list. The fewer terms you tag, the faster the tagging will be and the smaller your tagged database and glossary will be.

Note The exclusion applies to whole pieces of text, not to individual words within a piece of text. For example, a piece of static text "Notes is a product of Lotus Development Corporation" *will* be tagged if the exclusion list contains "Notes" and/or "Lotus". It will *not* be tagged if the exclusion list contains the whole phrase or "Notes*".

Exclusion list settings are stored in the Windows registry. You can find them in:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Lotus\Domino Global  
WorkBench\5.0.5\WorkBench\ExcludeTerms
```

Reviewing the results of tagging

When the tagging process is complete, you can review the results of the WorkBench process in the reports database. This will show you what was tagged.

In the glossary where the WorkBench put the new terms you should carry out a review. An important task is to identify pieces or text that should remain untranslated. You should select these in the All Terms with Translations view and then click Prevent Translation. See "Locating text to exclude from translation" in Chapter 5.

Building language databases

To build language database(s) you will need a tagged database and the glossary or glossaries referenced by it. There are three basic steps to building:

1. Defining the language databases.
2. Selecting the design elements to be built.
3. Creating the language databases.

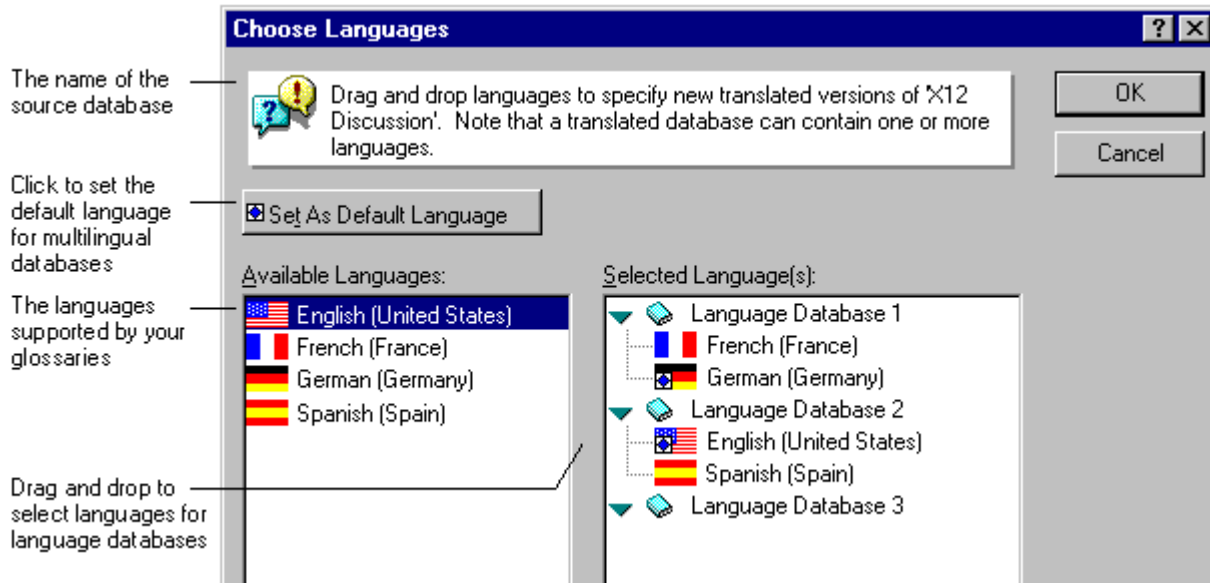
Note The language databases will not appear in translated (or pseudo-translated) form until you have translated the glossary or glossaries. See "Using the Translate/Review form" in Chapter 5 and "Creating pseudo-translations" in Chapter 4.

For information about language databases see "Language databases" in Chapter 1.

Defining language databases

Before you can create language databases, you must define them. The steps are:

1. In the WorkBench, highlight the tagged database.
2. Choose Actions - Choose languages, or click the Choose Languages button which appears if no language databases have yet been defined.
3. Complete the Choose Languages dialog box.



Notes You define each required language database by dragging languages to them in the dialog. You can define several output (language) databases in one operation. Each may be in one or more languages. If you choose several languages for a database, users will be able to choose which language they want to work in from Notes.

Initially, language databases have the name Language Database X, where X is a number. Only when the databases have been created will you see their full pathnames.

In a multilingual database, the default language is the language users will see if they select a language that is not available. Set the default language by highlighting a language and clicking Set as Default Language.

To change language selections for an existing language database, drag languages to or from it in this box. If you add a language to (or remove a language from) an existing built database, the text for that language will be added (or removed) when you update the database.

If you are running a pseudo-translation build, the language is 'Pseudo'.

Caution You may find updated information about multilingual databases in the README file in the Domino Global WorkBench program folder.



4. Click OK to return to the WorkBench display.

Caution The definition information for a language database is saved only when the database is actually created. If you leave the WorkBench, or load another project before creating a defined language database, the definition information will be lost.


Selecting design elements for building

Once you have defined language databases you must select elements for building. You can use the same process to change build selections for existing language databases before updating or rebuilding.

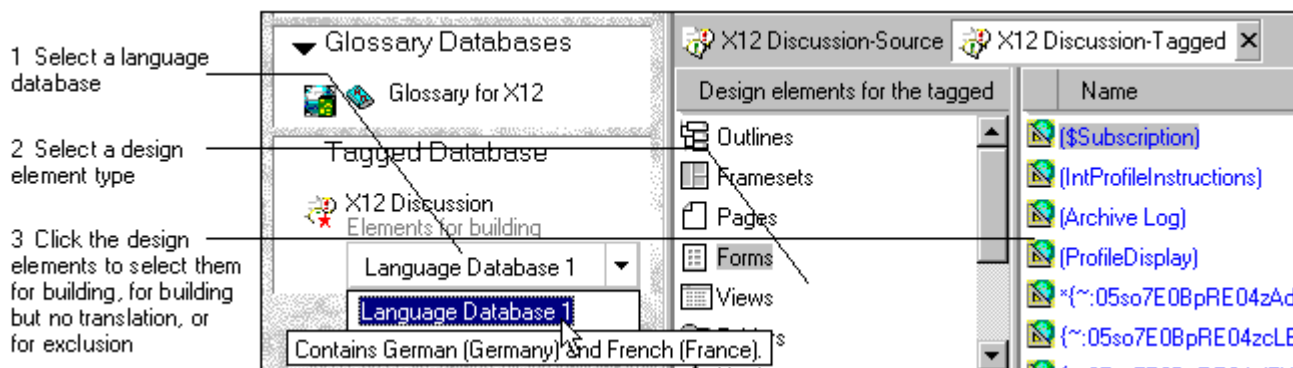
The steps are:

1. In the WorkBench, select one of your defined language databases in the drop-down box on the WorkBench display.
2. Select a design element type in the Design elements pane.
3. Make selections by marking each design element in the Name pane. The marking determines how the language database selected in the drop-down box will be built (or updated). Elements marked  will be translated (localized); elements marked  will be copied to the language database without translation; and unmarked elements will be excluded from the build process altogether. By default all elements are selected for translation.

If a term has been marked in the glossary as 'Prevent Translation' it will not appear translated in the language database even though it will have been through the full translation process.

During the building of language databases by the WorkBench, LotusScript is *only* recompiled if the design element that contains it has been selected for translation (marked ). In applications where script libraries are used it is important to consider that choosing not to build an element may cause problems because uncompiled scripts could contain translatable strings. This is particularly true if elements use a script library that contains string constants that are declared in the global declarations section. Also, remember that script libraries can use other script libraries so it is important to audit the script library usage structure before disabling elements from being built.

Right-clicking in the Name pane displays a menu that allows you to select or deselect all design elements of the current type, or open the properties box so that you can see information about a particular design element (for example its fields).



4. Repeat Steps 2 and 3 for each different design element type.
5. Repeat Steps 1 to 4 for each of your defined language databases.

Creating (building) language databases

Before you can create language databases you must have defined them and selected elements for building. Then, the steps for creating the database are:

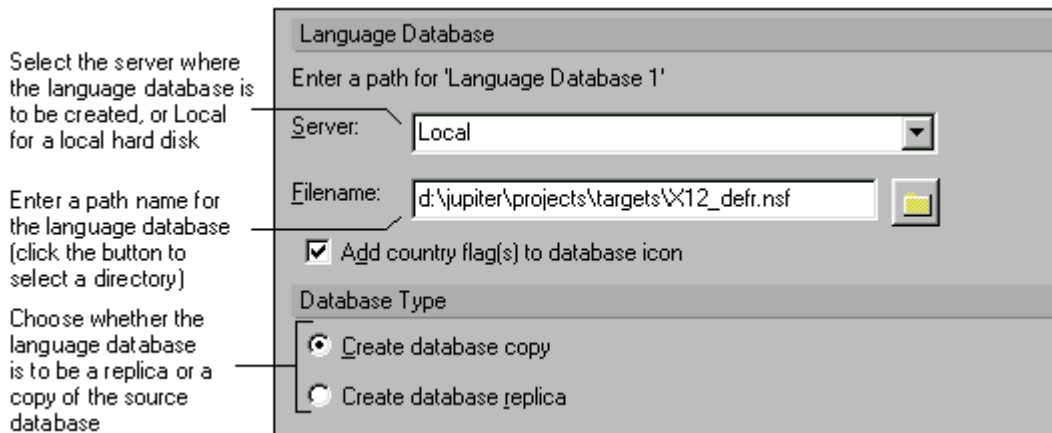
1. In the WorkBench, select one of your defined language databases in the drop-down box on the WorkBench display.



2. Click Build Language Database.

Note If the language database already exists you should either use Update Language Database to bring it into line with the tagged database, or use Rebuild Language Database to rebuild it. These options are available on the Tagged Database menu which appears when a tagged language database is highlighted.

3. Complete the New Database panel of the Language Database Creation dialog box.

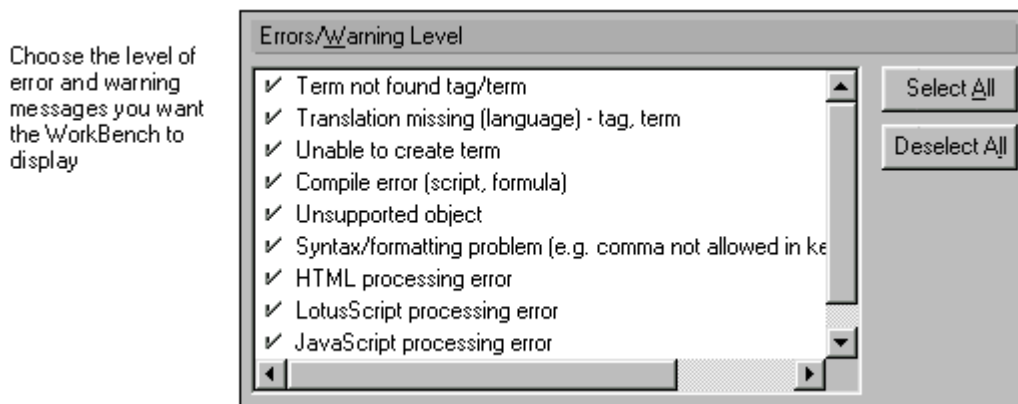


Notes A language database can be .NSF or .NTF format (the default is .NSF).

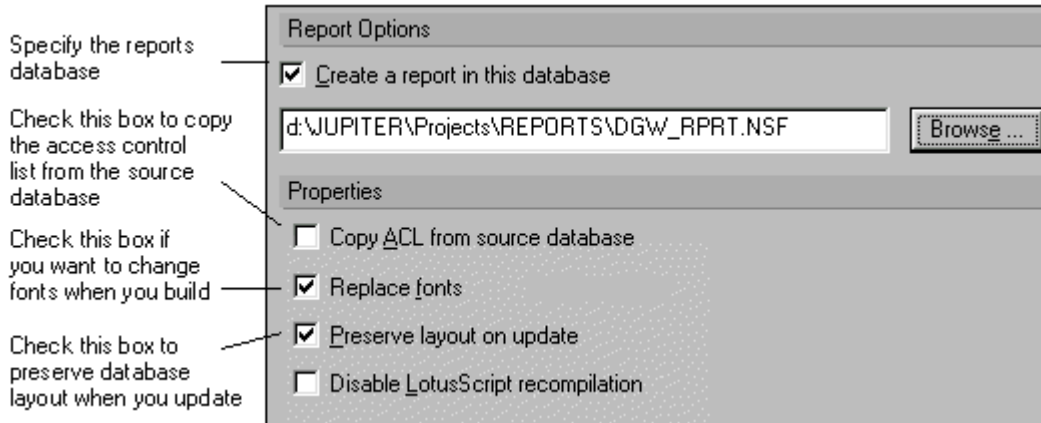
If you plan to synchronize language databases, you must put them all in the same directory.

Check 'Add country flag . . .' to add a country flag to the database icon. The WorkBench will use the flag appropriate to the database's language (the default language if it is a multilingual database).

4. Complete the Basics panel of the dialog box.



5. Complete the Options panel of the dialog box.



Notes To save a report of the build process, select a Report database. If you do not already have a report database you can create one by clicking Browse to open the Select Reports Database dialog and then clicking Create.

If you uncheck 'Create a report in this database' no report will be created, but you will still see results of the build process in the log pane at the bottom of the screen.

Choose Copy ACL to copy the access control list to the language database.

Caution If you are building a language database on a server, and the database includes a design element that has only reader access, always select 'Copy ACL from source' on the Build Options dialog. If you do not do this you will see unexpected errors (because the WorkBench does not see the reader access design element in the language database and, when you update the project, tries to add it a second time).

The 'Replace fonts' option is described below.

The 'Preserve layout' option only applies when you use the WorkBench to update or rebuild a language database. See "Preserve layout", under "Updating", below.

The 'Disable LotusScript recompilation' box is unchecked by default. So, for elements that have been selected for building and translation (🔧) on the Name pane, LotusScript is recompiled when you build. If you check this box, LotusScript is not recompiled when you build, even if you have chosen to localize LotusScript on the Tagging Options dialog and have selected the relevant design elements for building. Instead, the compiled script from the source database is copied unchanged to your language databases. It is the compiled LotusScript that gets executed in a Domino application. This option may be useful if:

- You have chosen not to tag LotusScript so there is no reason to recompile the existing scripts, or
- You have made other arrangements to recompile your LotusScript code, for example through the C-API or the IDE.

6. Click OK to build the database.

Caution If you cancel the building of a new language database (by clicking the stop icon at the top right of the display) and then close the project, the WorkBench will not be able to rebuild it later. This is because language information is stored in the icon note, and the icon note is not created until the end of the build process. If you want to create the language database you will have to delete the part-built version, and then re-define and build it again from scratch.

Note The WorkBench does not build translated About or Using documents into multilingual databases. See "Preparing a database for building multilingually" in Chapter 3.

7. View the log window or the report database and fix any problems. For example, if translations are missing add them to the appropriate glossaries. Rebuild if necessary.

Replace fonts

When the WorkBench builds a language database it uses, by default, the fonts that are used in the source database. However, if you check 'Replace fonts', fonts will be changed according to font mapping documents held in the project database, PROJECT.NSF in the Domino Global WorkBench program directory.

To create a font mapping document, open the project database, select the Font Mappings view and choose Create - Font Mapping. Specify the following:

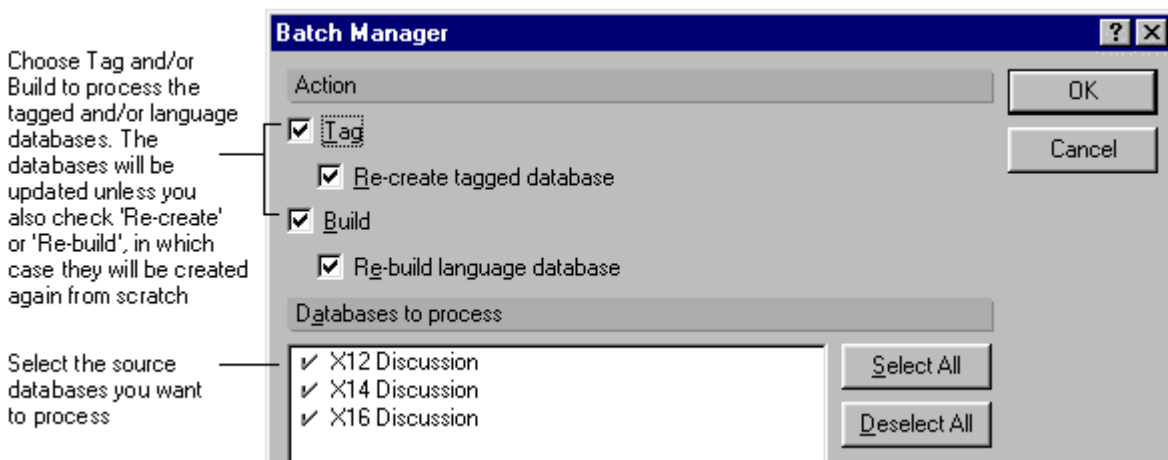
- The language that the font change applies to. If you want to apply the same font change to several languages, you must create a font mapping document for each language.
- The typeface and size of the font to be replaced (the font in the source database). You can enter an asterisk wildcard in the size field to represent all sizes.
- The typeface and size of the replacement font (the font to be used in the built language database). You can enter an asterisk wildcard in the size field to make the size the same as that of the font being replaced.

Font mapping documents have no effect if 'Replace fonts' is unchecked when you build.

Batch processing

You can update and/or rebuild several tagged and/or language databases for a project in one operation by running a batch process. To do this:

1. In WorkBench, open the project and choose Actions - Batch Processing.
2. Complete the Batch Manager dialog box.



3. Click OK. The updating and/or rebuilding operations will begin.

Note Any options which would cause prompts to be presented are disabled for the duration of the batch process. For example, if a source database had the tagging option 'Always Prompt' then this would be disabled for the batch run on that source database. The other existing tagging and build options set for each source database are honored.

The modification of the prompt settings for batch processing is only temporary. The original settings are recovered if the tagged database is updated or rebuilt in 'normal' mode. The temporary changes made to the options for the batch process are written to the report.

Updating

If the design of a source database changes you will need to update its tagged database and associated language database(s). The changes to the source database may have been received by replication, or the new version of the database may have been copied over the existing one.

There two basic ways of handling updates:

1. If there have been relatively few changes you can use the WorkBench update function to bring the tagged and language databases into line with the new source. Some new glossary entries will require translation. This is an automatic process:
 - The WorkBench looks at the date stamp of design elements in the databases to determine which of them have changed. Color coding on the WorkBench display shows the results of this pass. In general WorkBench does this automatically, but if you change a source database while WorkBench is running you should choose View - Refresh to ensure the change information is up to date.
 - When you run the update, WorkBench rebuilds the design elements that have changed. For forms, it rebuilds only the design fields that have changed (for example \$Title and \$Body).

Note The layout and static text of a form is held in a single field called \$Body. So if a piece of static text in a form changes, the layout and all the static text in the form will be rebuilt on update.

As part of the update process new text is tagged and added to the glossaries as necessary. In this way glossaries are automatically updated. Of course any new text will need translating.

A drawback of this approach is that the update function *appends* any new terms the end of the glossary, so the translators cannot easily see them in context.

See "Modifying source databases after localization has started" in Chapter 3.

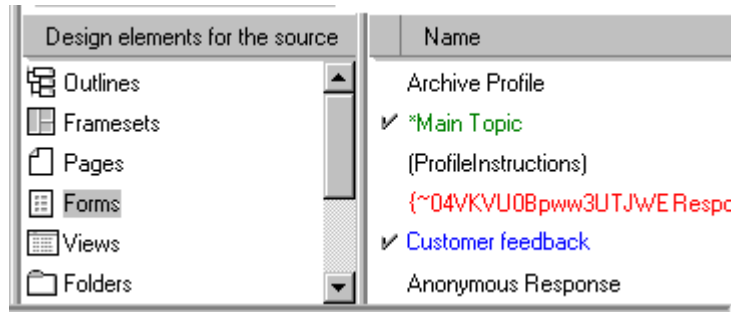
2. If there have been many changes it may be better to start from scratch, creating a new glossary and a new tagged database. There are two reasons for this:
 - Although the update function (above) creates new glossary entries for new text, it does not remove glossary entries for obsolete text. So, if the changes are extensive, you may find your glossary cluttered with a lot of redundant entries.
 - Terms are tagged in the order in which the WorkBench finds them, which makes it possible, in a newly filled glossary, to list them in context order rather than alphabetical order. Translators click the Tagged Order Sorting button on the Translate/Review dialog to see terms in context order. Therefore, if there have been a lot of changes, translation can be quicker and more accurate with a new glossary.

A drawback of this approach is that the new glossary initially contains no translations. You will have to use Export, Import and/or Retrieve to recover comments and DNT information from the previous, translated, glossary.

Using the WorkBench to update projects

To update tagged and language database(s):

1. Select the source database. If it is already selected, choose View - Refresh to ensure that the latest changes are reflected in the WorkBench display.
2. Review the changes by selecting each design element type in turn (in the 'Design elements' box). In the 'Name' pane, color coding shows the differences between the source and tagged database.



Elements shown in black are the same in source and tagged databases (they have the same date and time stamp); elements shown in blue are present in the source, but not in the tagged database; elements shown in red are present in the tagged database, but not in the source database; and elements shown in green are in both source and tagged databases, but they are not the same (they have different date and time stamps).

To see information about a particular design element (for example its fields) right click it in the 'Name' pane to open its properties box.

Notes If a design element has been re-created (for example by copying and pasting), rather than just edited, it will have a different unique ID and WorkBench will treat it as a new design element. In the WorkBench 'Name' pane it will be displayed in blue in the source database and red in the tagged database. However, if a source database is replaced by a copy or replica of itself its design elements will retain their original unique IDs (UNIDs), and updating will work correctly.

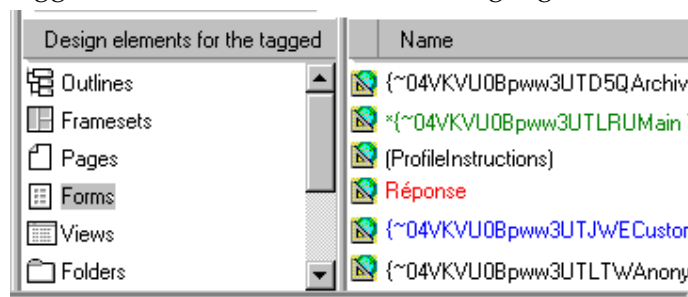
Forms that have not apparently changed can appear in green (changed) if the source database is an authorized Notes template. In that case, any design change in the source triggers a warning that its authorization will be invalidated if you proceed. If you click Yes, Notes opens every form and removes its \$Licensee field. Such a change is correctly picked up by the WorkBench. The \$Licensee fields can be removed only once, so any subsequent design changes will be displayed as expected.

3. Make tagging selections in the 'Name' pane for new (blue) and changed (green) design elements. Selections you make for other elements have no effect: elements that are present only in the tagged database (red) are always removed on updating, and design elements that have not changed (black) are always excluded from the update process.



Note Design elements that have changed, but which do not require tagging or retagging, are shown in black, like unchanged design elements. They are copied through to the tagged database, and from there to the language databases, transparently.

See "Selecting elements for updating", below.

4. (Optional) Choose Actions - Set Tagging Options to open the Tagging Options dialog and adjust the tagging options.
See "Creating the tagged database", above.
5. Click Update Tagged Database (or, with the source database selected, choose Source Database - Update Tagged Database). The WorkBench will now bring the tagged database into line with the source database. Unless you specified tagging to be automatic, you will see the Tagging Window when the WorkBench finds new or changed text in the source database.
See "Creating the tagged database", above.
6. Highlight the tagged database and select one of the language databases in the drop-down box.
7. Review the changes by selecting each design element type in turn (in the 'Design elements' box). In the 'Name' pane, color coding shows the differences between the tagged database and the selected language database.



Elements shown in black have the same date and time stamp in both tagged and the language database; elements shown in blue are present in the tagged, but not in the language database; elements shown in red are present in the language database, but not in the tagged database; and elements shown in green are present in both tagged and language database, but they have different date and time stamps.

8. (Optional) Adjust the build selections in the 'Name' pane. Elements marked  will be translated; elements marked  will be copied to the language database without translation; and unmarked elements will be excluded from the build process altogether.

Design elements that have changed, but which do not require tagging or retagging, are shown in black, like unchanged design elements. They are copied through to the tagged database, and from there to the language databases, transparently. Design elements shown in red are elements that have been deleted in the source and tagged databases, and therefore they cannot be selected here. They will be removed from the language database.

See "Selecting elements for updating", below.

9. (Optional) Adjust the options the Build Options dialog.
Note When updating it is usually a good idea to ensure that 'Preserve layout' (described below) is checked.
10. Click Update Language Database (or, with the tagged database selected, choose Tagged Database - Update Language Database). The WorkBench will now update the language database.
11. If necessary, repeat Steps 6 to 10 for other language databases.

12. If your language database(s) are synchronized databases, you will now need to update their language synchronization features. Go to the language synchronization display to do this (see "Adding language synchronization features" in Chapter 8).

Preserve layout

When the WorkBench builds a language database it uses, by default, the layout (table cell widths, column widths in views, field positions in layout regions, and so on) from the *source* database. After building, it is usual to have to manually adjust layout to take account of such things as text expansion. See "Design changes in language databases" in Chapter 7.

With the 'Preserve layout' option on the Build Options dialog checked, the WorkBench saves as much layout information as possible from the *language* database, and then reapplies it once the updating operation is complete. Selecting 'Preserve layout' therefore eliminates much of the need for reworking of layout when databases are updated.

The 'Preserve layout' option has no effect when you build a language database for the first time.

Note The 'Preserve layout' option operates for rebuilding as well as updating operations. To prevent layout from being taken from the language database, uncheck 'Preserve layout' on the Build Options dialog before you update or rebuild.

An option, 'Recover Layout from Database', on the Language Database menu, lets you apply layout settings from another database to the currently selected language database. Usually the other database will be an earlier version of the current one, perhaps built by a previous version of Domino Global WorkBench.

Selecting elements for updating

When a source database has changed, and you select or deselect elements for tagging and/or building, this is what the update procedure does:

- When updating the tagged database

Element status	If element selected for tagging	If element not selected for tagging
Unchanged (black)	No action. Existing element in tagged database unchanged.	No action. Existing element in tagged database unchanged.
New (blue)	New element copied from source database and tagged.	New element copied from source database, but not tagged.
Deleted (red)	Deleted element removed from tagged database.	Deleted element removed from tagged database.
Changed (green)	Changed element copied from source database and tagged.	Changed element copied from source database, but not tagged.

- When updating a language database

Element status	Selected for translation	Selected for no translation	Not selected
Unchanged (black)	No action. Existing element in language database unchanged.	No action. Existing element in language database unchanged.	Existing element in language database deleted.
New (blue)	New element built from tagged database and translated.	New element built from tagged database, but not translated.	No action. Existing language database unchanged.
Deleted (red)	Deleted element removed from language database.	Deleted element removed from language database.	Deleted element removed from language database.
Changed (green)	Changed element built from tagged database and translated.	Changed element built from tagged database, but not translated.	Remove from language database.

Forcing an update

The WorkBench updating process only operates on design elements (such as forms and views) that have *changed* in some way, in other words they have been edited, added or deleted. The update process will not operate:

- If the only change has been to the selection or deselection of elements for tagging or building.
- If the only change has been to an entry or entries in a glossary.

However, you may want to run an update in these circumstances. For example, on reviewing built databases you may decide to make changes to translations or to the 'Prevent Translation' marks in a glossary. Rather than rebuilding entire language databases (and losing formatting work done on them) you may prefer to run an update to rebuild only the elements that are affected by the changes.

To force an update in these circumstances:

1. In the WorkBench, right-click the source database and choose Open in Designer.
2. Open a design element that you want to update.
3. Make a trivial edit to it. For example, add a space to a piece of text.
4. Save the document.
5. Repeat Steps 2 to 4 for any other elements you want to update.
6. Close the database and return to the WorkBench.
7. Choose View - Refresh. You should now see the elements you edited marked as changed (colored green). They can now be updated in the usual way.

Rebuilding databases

There are circumstances when you might want to rebuild (rather than update) a tagged or language database. For example:

1. If you decided to change the selection of elements for tagging or building.
2. If you want to change tagging or building options.
3. If you want to add or remove languages in a multilingual database.
4. If you have made changes to translations in one or more of the glossaries.

When you *rebuild* a database, the WorkBench re-creates it entirely from scratch, overwriting the original database file. Consequently, any changes previously made in the database are lost. On the other hand, when you *update* a database, the WorkBench only processes those elements that are new or have changed, and necessary changes are made to the existing database file. Any modifications previously made in the database (other than to elements that are now being updated) are preserved.

Note The 'Preserve layout' option on the Build Options dialog operates for rebuilding as well as updating operations. To prevent layout from being taken from the language database, uncheck 'Preserve layout' before you update or rebuild.

Rebuilding a tagged database

The procedure is:

1. In WorkBench, highlight the source database.
2. (Optional) Adjust tagging selections in the Name pane.

3. (Optional) Choose Actions - Set Tagging Options to adjust tagging options.
Note If you omit this step, the WorkBench will use the existing tagging options.
4. Choose Source Database - Re-create Tagged Database (or click Re-create Tagged Database).

Rebuilding a language database

The procedure is:

1. In WorkBench, highlight the tagged database.
2. In the drop-down box, under the tagged database icon, select the language database you want to rebuild.
3. (Optional) Adjust build selections in the Name pane.
4. (Optional) Choose Actions - Choose Languages to add or remove languages.
5. (Optional) Choose Actions - Set Building Options to adjust build options.
Note If you omit this step, the WorkBench will use the existing build options.
6. Choose Tagged Database - Rebuild Language Database (or click Rebuild Language Database).
7. If the database was a synchronized database, go to the language synchronization display and add the language synchronization features.

Tagging

Tagging is the process of linking database text to entries in a glossary, and it is an essential part of the creation of a tagged database. Each piece of text in the tagged database is 'tagged' with an ID that uniquely identifies an entry in a glossary. Associated with the glossary entry (once the glossary has been translated, see Chapter 5) are translations for the term in whatever languages are needed.

In general tags are created automatically by the WorkBench when you create a tagged database. At the same time, the WorkBench can create new glossary entries for any unfound text.

The WorkBench will correctly tag most pieces of text. However, some text you may not want to tag, for example text that is never seen by users of the database. When creating a tagged database with WorkBench you can exclude text from the tagging process at several levels:

- In the WorkBench main display you can deselect elements in the source database. These elements will not be tagged. They will pass through to the tagged database unchanged.
- In the Tagging panel of the Tagging Options dialog box you can exclude specific types of text (for example HTML or LotusScript) and specific words from the tagging process.
- In the Prompting panel of the Tagging Options dialog box you can specify that the WorkBench should prompt you before tagging text in formulas. This is useful because it is not always clear from the context whether text between quotes in a formula should be tagged or not. In general, if it is visible to users it should be tagged, otherwise not.

Notes Rather than excluding text from the tagging process, it is often simpler to let WorkBench tag all text in the database and then, in the glossary, mark as 'Prevent Translation' any text that is never displayed, or must remain in the reference language for functionality reasons.

The WorkBench's automatic tagging will be much more accurate if everything in the source database has been aliased, and a standard system of prefixes has been adopted. See "Using aliases" in Chapter 3.

The following sections explain what text should and should not be tagged (any untagged text passes unchanged through to the language databases).

Tags

Tags in a tagged database look like this:



In this example several form names are tagged. The tag delimiters and IDs are regular text: if you need to remove a tag, open the tagged database in Notes and delete the delimiters and ID.

The first 11 characters of a tag are generated from the replica ID of the glossary where the term is stored. The next 6 characters are the unique identifier for the term.

Before you create a tagged database you must choose the delimiters you want to use. You do this on the Tagging panel of the WorkBench's Tagging Options dialog. Choose combinations of characters that do not appear anywhere in your database. Use at least two characters for the start and end delimiter. The defaults are {~ and ~}. In Lotus databases, ^ and {^ work well as alternative opening and closing delimiters. Do not use '[' or ']' as tag delimiters because Domino uses them to mark HTML tags.

Note The open and close brace characters, { and }, are used as literal string delimiters by LotusScript. If you use these characters in tag delimiters for the WorkBench, you may see some anomalies when you view the tagged database. However, this will not cause problems in the built language databases.

Tagging in general

You should tag for translation all text that is visible to users. And you may also want to tag (though not translate) text that is not visible to users. In particular:

- **Static text** Static text on forms should be tagged for translation. The WorkBench tags static text.
- **Design element names** As explained in Chapter 3, you should create aliases (synonyms) for the names of all agents, folders, forms, views, navigators and keywords, and then:

Tag the name

Do not tag the alias

Where aliases are indicated by vertical bar characters it is the *first* (leftmost) segment that is the display name, for example: Name | Alias1 | Alias2. Aliasing ensures that Notes functionality is not disturbed in cases where an item name is used in (for example) a selection formula or LotusScript. Notes functionality will use the alias, and the name itself (which may display) can then safely be tagged.

The WorkBench tags all design element names, except for field names and subform names, and it does not tag aliases (these items should never be translated).

- **Text that will *not* be seen by database users**, for example the names of hidden design elements, or comments. You have two choices, either:

Tag it, but mark the glossary entries 'Prevent Translation'. This is recommended because you will not need to consider the text again if you create another tagged database using the glossary: any pieces of text that should not be translated will be connected automatically with their appropriate 'Prevent Translation' entries in the glossary.

or

Leave it untagged. It will then be copied unchanged to the built (language) databases. However, each time you create a tagged database that contains the text it will show up as a 'new' term in your glossary, unless you take care to exclude it from the process. When a database is in development you may need to create tagged databases quite frequently.

Except for aliases, the WorkBench tags most text not seen by users.

- **References to design element aliases in formulas or LotusScript** There are two ways to handle these:

If you have used standard alias prefixes in your source database, you can make the WorkBench skip them automatically when it creates the tagged database. In that case they will always remain in the reference language (as is necessary) and will never result in glossary entries.

Otherwise, these references can be treated as described in the previous section: it is then recommended that you tag them, but mark the corresponding glossary entries 'Prevent Translation'. See "Tagging references to other design elements".

The WorkBench tags references to aliases in formulas.

- **References to design element names in formulas** Sometimes you may encounter references in formulas to design element names, not aliases. Usually the design element name will have been tagged, in which case the reference in the formula will need to be tagged for translation as well. See "Tagging references to other design elements".

The WorkBench tags references to design element names in formulas.

Do not tag any text unit bigger than a paragraph. If you have several contiguous paragraphs, tag each one (or, better, each sentence) separately.

Keep in mind the needs of other languages. It is important to have separate tags and glossary entries for instances of text that, though the same in English, might be translated differently. For example in many languages there are gender agreements or matters of conjugation to pay attention to (not usually an issue in English). Multiple tagging will allow translators the necessary flexibility to produce proper language versions. It's a good idea to get the advice of a translator in advance about which pieces of duplicated text will need separate tags. If you select all of the 'Context Matching' options on the Prompting panel of the Tagging Options dialog, you will get separate tags and glossary entries automatically.

If you have to tag text in formulas, be consistent. For example, in @Command([OpenNavigator];"(Home)";"0") you could tag the string Home or the string (Home). In this case, tagging the whole string, with the parentheses, will result in a glossary entry that can easily be identified as one that needs to be marked 'Prevent Translation'.

Tagging references to other design elements

In Notes, there are two basic ways of referring to other design elements: in formulas and through Notes built-in functions.

In formulas or LotusScript

References to other design elements are common in formulas and in LotusScript. In such cases the text in the formula can reference either:

- The *alias* for the element name (recommended), in which case it should not be translated
or
- The *name* of the element itself, in which case it should be translated.

See "Formulas" and "LotusScript", below.

Through Notes built-in functions

References to other design elements are also common in Notes built-in functions, such as simple actions. Here you choose a design element from a list. However, the list shows only the element *names*, not their aliases (if any), and there is no way to change a selected list entry to make it reference an alias rather than an actual name.



This is part of the design pane for a navigator. A view is selected by means of a simple action. The choices presented in this box are the actual names of the views in the database. Aliases are not listed. And there is no way of editing the entry to refer to the alias rather than the view name itself. The view name will normally be tagged for translation, so WorkBench also tags this text. It does so by using the Notes API

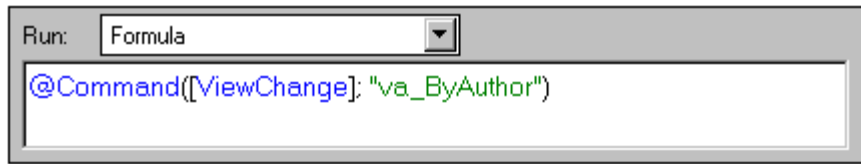
The WorkBench tags the list entry by using the Notes API. This means that:

- If the name of the referenced design element is also tagged, and if it is tagged to the same glossary entry, there will be no functionality problems in the language databases.

Note The WorkBench can only make tags to the same glossary entry if you do not select any of the 'Match the following on lookup' options (other than Database title) on the lookup options panel.

- If the name of the referenced design element is also tagged, but it is tagged to a different glossary entry, there will still be no problems with the functionality in the built language databases *provided* the two glossary entries are translated the same way.
- If the name of the referenced design element is not tagged, then functionality in the language databases will be broken. To fix this problem, tag the design element name and then select it in the simple action box.

One way to avoid these potential problems is to use a formula instead of a built-in simple action. The formula can then be handled in the usual way.



Here a formula is used instead of the simple action shown in the previous illustration

Handling attributed text (rich text)

In order to optimize storage space, Notes breaks text at attribute changes and stores the segments of text separately. So the text "the **black** dog" is held as three separate pieces of text. The segmentation remains even if the attribute change, the bolding of the word "black", is subsequently removed. (This breaking of text does not affect what you see on the screen, it is simply an efficient method of storing text.)

When working with plain text, the WorkBench, by default, follows the Notes segmentation behavior. So, in the example above, you would get three separate glossary entries. (This is how Notes Global Designer 4.6 handled the tagging of attributed text.)

In Domino Global WorkBench, however, you can change this behavior. If you check 'Combine Broken Terms' on the Lookup panel of the Tagging Options dialog:

The WorkBench will handle text as complete strings wherever it can. In particular, it will combine text that is broken because it once contained an attribute change, but does no longer. And it will combine text that is broken because it contains an invisible attribute change, for example bolding on a space character.

However, where there is a real, visible attribute change in the text string, the string will still be segmented in order to preserve the attributes of the separate sections of the text.

Each piece of text will be matched only against plain text glossary entries, and any glossary entries that are created will be plain text ones.

If you also check 'Rich Text Sensitive':

Pieces of text that contain visible attribute changes (such as bolding, italicizing or font changes) will be handled as complete units.

A piece of text that contains a visible attribute change will be matched only against rich text glossary entries, and any glossary entries that are created from such text will be rich text ones. A piece of text that contains no visible attribute change will be matched only against plain text glossary entries, and any glossary entries that are created will be plain text ones.

Note Rich text glossary entries are new in Domino Global Designer Release 5.

You should normally leave 'Combine Broken Terms' and 'Rich Text Sensitive' checked. However, if you are using a glossary from Notes Global Designer Release 4.6, which did not handle rich text, you may want to uncheck these options in order to improve the chances of finding matches. In Release 4.6 any attribute changes, including invisible ones, would have led to segmented glossary entries.

When the WorkBench looks for matching rich text entries in glossaries, it looks for an *exact* match: the attributes must match as well as the text. Like visible characters, spaces can have attributes. Although they are not visible, differences in attributes on spaces will affect the results of WorkBench's matching operations.

Certain glossary functions do not operate on rich text entries. See "Rich text terms" in Chapter 4.

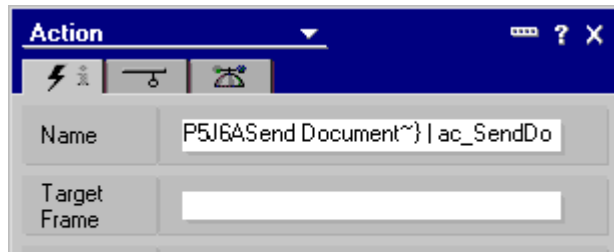
Tagging specific design elements

The following sections show you how to handle standard Notes design elements.

Note The WorkBench tags elements unless otherwise stated.

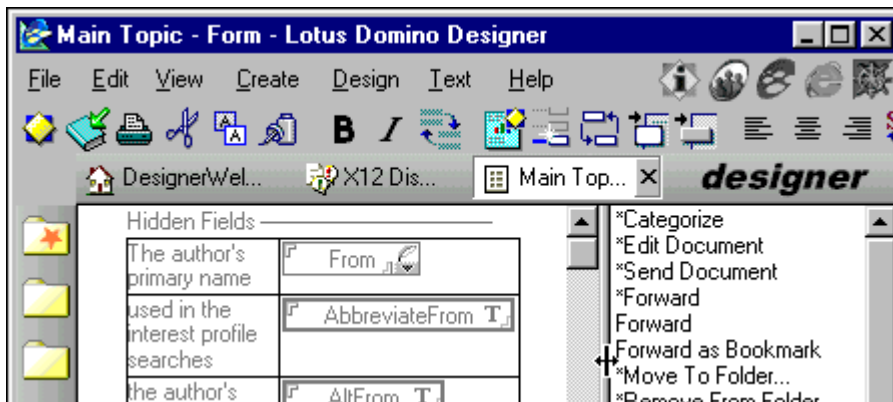
Actions

Action names are visible to users on buttons and in the Action menu, so they should be tagged for translation. They can be used for computation, so they should be aliased. In the tagged database you will see something like this:



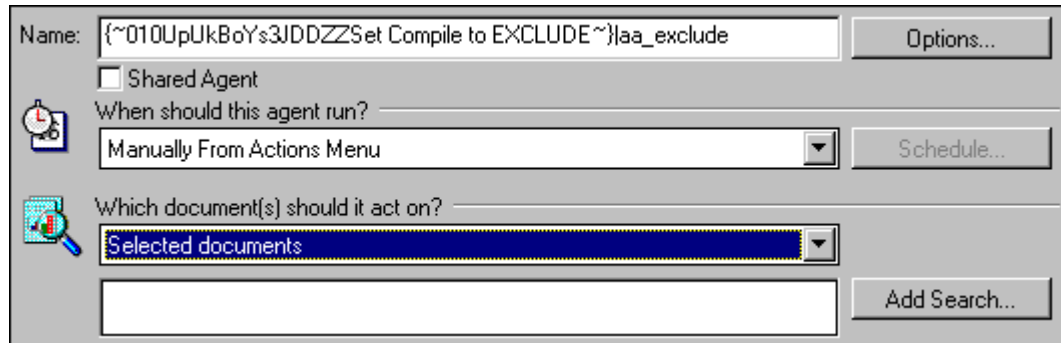
Note Because action names can appear in the Action menu, they can include hit letters, denoted in the usual way with an underscore, for example 'Check for _duplicates'.

Actions can be found on forms and in views. You drag the right hand edge of the design pane to reveal them:



Agents

Agent names can be visible to users, for example on the Action menu, so they should be tagged for translation. They can also be used for computation, for example they can be referenced by other agents, so they should be aliased. In the tagged database you will see something like this:



Comments

The purpose of comments is to help designers understand the database. They do not display to the user, so you will not normally want to translate them:

- You can tag them, but mark the glossary entry 'Prevent Translation'. This is recommended because you will not need to consider the comment again if you have to tag an updated version of the database.
or
- You can leave them unaliased and untagged.

The WorkBench does not tag REM statements.

Database titles

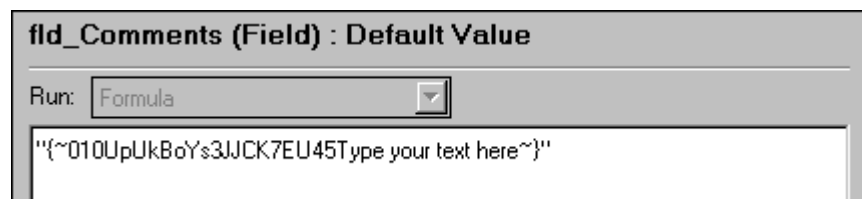
Database titles are displayed on the screen, for example in the desktop icon. You may or may not wish to translate them, it is really a matter of your own house style. You can tag them for translation, or, if you decide not to translate them, handle them in the same way as Comments.

Fields

Field names are never displayed on the screen, so they should not be tagged. The WorkBench does *not* tag field names. However, there are various places where you might find translatable text associated with fields. These are described in the following subsections.

Default value

A field can be given a default value. This will be visible to users, and so, if it is text, it should be tagged for translation. It is not used for computation, so it does not need to be aliased.



If the field is a keyword field, the reference here should be to an alias, rather than to displayed text, in which case it can either be left untagged or tagged and marked 'Prevent Translation' in the glossary.

Help description text

A field can have a Help description. This will be visible to users, and should therefore be tagged for translation. It is not used for computation, so it does not need to be aliased. In the tagged database you will see something like this:



If the field is a hidden field, any Help text will not display. Therefore:

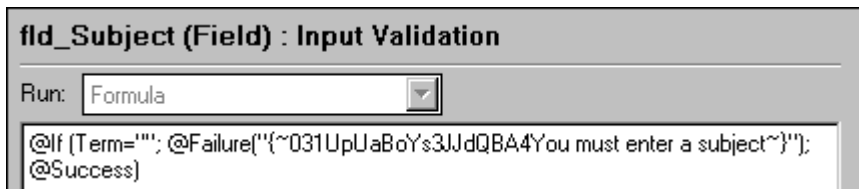
- You can tag it, but mark the glossary entry 'Prevent Translation'. This is recommended because you will not need to consider the text again if you have to tag an updated version of the database.
- or
- You can leave it untagged.

Input translation

A field can have an input translation formula. This may contain text that will be visible to users, which should therefore be tagged for translation.

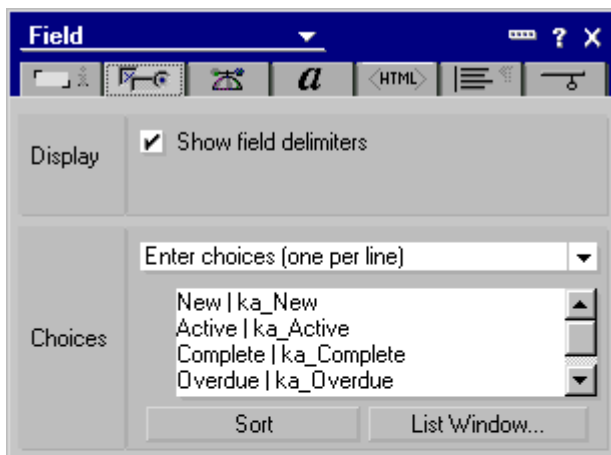
Input validation

A field can have an input validation formula. This may contain text that will be visible to users, which should therefore be tagged for translation. In the tagged database you will see something like this:



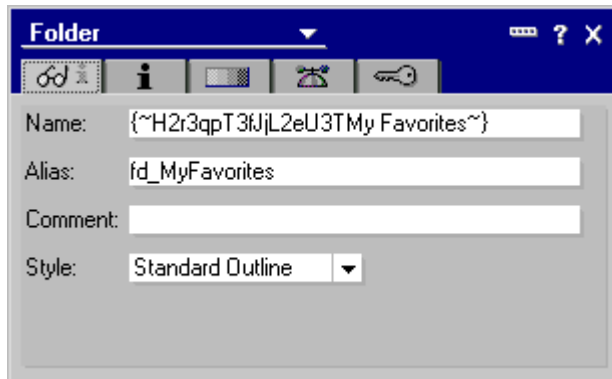
Keywords

Keywords are visible to users, so they should be tagged for translation. They can also be used for computation, for example they can be referenced in formulas, so they should also be aliased:



Folders

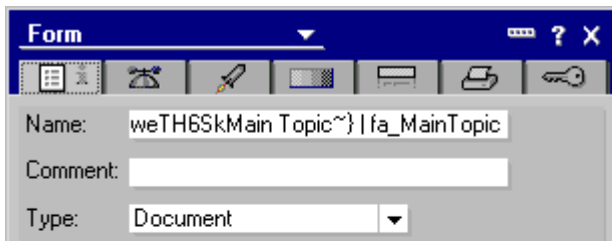
Folder names can be visible to users, so they should be tagged for translation. They can also be used for computation, for example they can be referenced in formulas, so they should also be aliased. In the tagged database you will see something like this:



Caution Do not translate the following special folder names: (\$Alarms), (\$All), (\$Calendar), (\$Drafts), (\$Inbox), (\$Meetings), (\$Profiles), (\$Sent), (\$ToDo), and (\$Trash). These appear in mail databases and Notes references them directly. Notes will be unable to find them if they are changed.

Forms

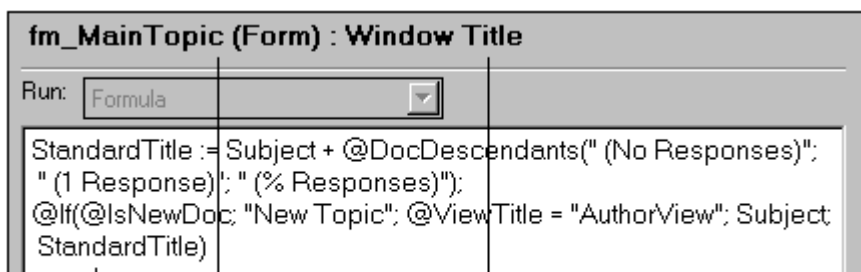
Form names can be visible to users, for example on the Create menu, so they should be tagged for translation. They can also be used for computation, for example they can be referenced in formulas, so they should be aliased. In the tagged database you will see something like this:



Caution Do not translate special form names that begin "\$\$", for example \$\$ViewTemplateDefault and \$\$ViewTemplate.

Formulas

Formulas can be short and simple or long and complicated, and they can be used in almost any part of a database's design. You will find them in the design pane for the various design elements:



This formula . . . in this part of the design . . . is associated with this event

Here are some tagging guidelines:

- Anything outside double quotes does not need to be tagged.
- *In general*, text inside double quotes can appear on the screen and needs to be tagged. This is what the WorkBench assumes.

However, text inside double quotes is often an alias referencing a design element or a dynamic variable, and in these cases it must not be translated. Either do not tag it, or tag it and mark the glossary entry 'Prevent Translation'. See "Tagging references to other design elements", above. The best way of overcoming this problem is to use a system of unique prefixes to identify aliases (see "Using aliases" in Chapter 3). You can then either:

Make the WorkBench ignore them when it creates the tagged database by making a wildcard entry on the Tagging panel of the Tagging Options dialog.

or

Quickly identify them in the glossary and mark them 'Prevent Translation'.

If you do not adopt a system of prefixes for aliases, there is no simple way of determining whether a reference in a formula is to a design element name or an alias. A design synopsis (see "Creating a design synopsis" in Chapter 3), which lists both element names and their aliases, can help. And functionality problems in built (language) databases can show where an alias has been translated by mistake.

See "Terms that must not (or need not) be translated" in Chapter 5.

The WorkBench disregards anything inside double quotes if it comprises only numbers or punctuation marks.

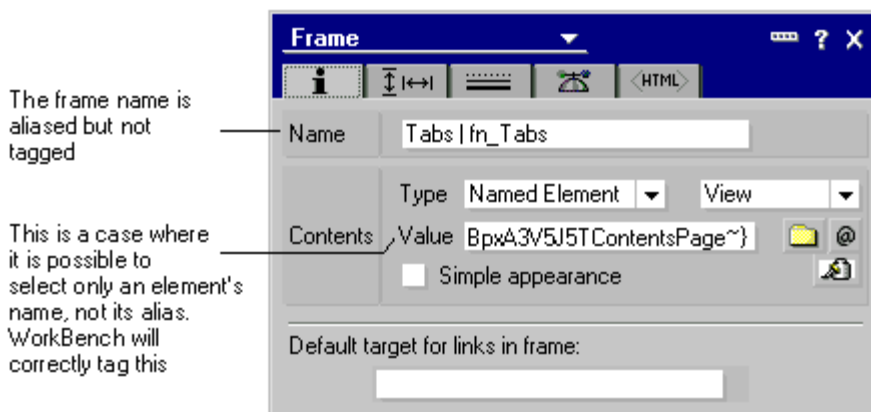
If you want to prevent the tagging of formulas, you can do so in the Tag the Following list box on the Tagging Options dialog.

Framesets and frames

Frameset names are not visible to users so they do not need to be tagged for translation. They can be used for computation, for example they can be referenced in formulas, so they should be aliased. By default the WorkBench does not tag frameset names.

Frames

Frame names are not visible to users so they do not need to be tagged for translation. They can be used for computation, for example they can be referenced in other frames, so they should be aliased. By default the WorkBench does not tag frame names.

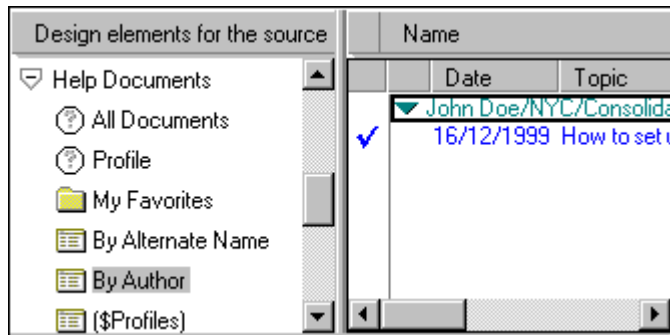


The 'Value' is either a formula or the name of another design element, for example a page or a form. In the latter case, the design element is chosen from a drop-down which only lists element names, not aliases. The WorkBench tags the 'Value' entry by using the Notes API.

The 'Default target . . .' is the name of another frame.

Help documents

Help documents are normal data documents that are part of a database's design (they are called 'help' documents because they usually contain information or instructions for database users). Unlike other design elements, the WorkBench does not, by default, select them for translation. But if you need to you can select them the WorkBench in a Notes-like display:



Note The 'All Documents' view of help documents is useful only if the documents contain a suitable predefined field. See "Help documents" in Chapter 3.

Hidden design elements

Hidden design elements are design elements that are not presented to the user for selection. Their names are shown in parentheses, are not seen by the user and do not need to be translated. Their contents, however, usually are visible to the user and should be tagged in the same way as they would be in a normal, unhidden, design element.

For the *names* of hidden elements:

- You can tag them, but mark the glossary entry 'Prevent Translation'. This is recommended because you will not need to consider the text again if you have to tag an updated version of the database.

Note A hidden design element will not appear as hidden in the tagged database because the WorkBench tags the whole name, including the parentheses. This prevents Notes from recognizing it as hidden. In built databases, however, the parentheses will be restored, so the element will again be hidden.

or

- You can leave them unaliased and untagged.

Note If you want to include hidden elements in the tagging process you must select them on the Tagging panel of the Tagging Options dialog.

HTML

Domino Global WorkBench handles HTML as follows:

1. Processed as HTML (only translatable text is tagged):

- Text marked as pass-thru HTML.
- Text in pages marked 'Treat contents as HTML'.
- HTML contained in fields named 'HTML' (see "Storing HTML in a field" in the Domino Designer documentation). The default formula for such a field is tagged as HTML only if it comprises a single quoted string, otherwise it is tagged as plain text.
- Rich text fields which are flagged 'Treat contents as MIME/HTML'. The WorkBench tags text where the content-type is text/html.
- The contents of forms that are marked 'Treat form contents as HTML'. The default formula for any field in such a form is tagged as HTML only if it comprises a single quoted string, otherwise it is tagged as plain text.

The tagging in Release 5.0.5 of the WorkBench has been enhanced so that it is better able to recognize HTML fragments, but their unambiguous recognition is not always possible in which case they will be tagged as regular text.

Domino Global WorkBench does not support some elements embedded in an HTML stream, such as bitmaps. Fields and table markers are supported, but HTML that includes other embedded elements will have to be excluded from the tagging process (or marked 'Prevent Translation' in the glossary) and localized 'manually' in each output database, after building.

You may find that there are translatable values for some parameters that the WorkBench does not tag. You will have to translate these directly in the built language databases.

2. Processed as normal text (all text, including HTML coding, is tagged):

- HTML in paragraphs that have been given the paragraph style 'HTML'.
- HTML embedded in square brackets.
- HTML (or pieces of HTML) in formulas.
- HTML in a view or folder where 'Treat text as HTML' is checked.

Note If there's a design element property enabling HTML, that will take precedence over a pass-thru property of any type. So, for example, text in square brackets in a page marked 'Treat contents as HTML' *is* tagged.

If you want to prevent the tagging of HTML, you can do so in the Tag the Following list box on the Tagging Options dialog.

Java

By default Domino Global WorkBench Release 5.0.5 tags text in Java .properties files. In a Domino database, .properties files are stored in individual design elements, where they form part of a Java agent or applet, or as shared applet resources. The WorkBench tags Java text even when the .properties files are held in a Java archive files (.JAR or .ZIP). An entry in a .properties file has the format key=value. The 'value' is the text that may need translation and which is tagged.

Java text can be excluded from the tagging process in the following ways:

- By removing the tick from the Java entry in the Tag the Following box on the tagging panel of the Tagging Options dialog box.
- By adding individual text entries to the Exclusion List on the tagging panel of the Tagging Options dialog box.
- In the case of shared applet resources, by deselecting individual applet resources on the main screen prior to tagging.

Usually it is advisable to tag all Java properties text and then, in the glossary, to mark any strings that do not need translation using the 'Prevent translation' button.

JavaScript

JavaScript is processed by the WorkBench in the following situations:

- When it appears between `<script>` and `</script>` tags in any HTML that is processed by the WorkBench (listed in Item 1 of "HTML", above).
- Wherever native JavaScript is supported in Domino Designer. For example in views, in events or in actions placed on the action bar.


In these circumstances text strings are tagged for translation. A 'string' is any text that appears between double or single quotes.

You should exclude functional strings from the tagging operation, for example strings that identify browsers ("Netscape") and HTML generators ("Mozilla"). Enter these in the Exclusion List on the Tagging panel of the Tagging Options dialog. If you want to prevent the tagging of JavaScript, you can do so in the Tag the Following list box on the same panel of the Tagging Options dialog.

LotusScript

The WorkBench handles LotusScript in much the same way as it does formulas: it regards text in quotation marks as translatable, with the exception of text that is part of the LotusScript language itself. See "Formulas", above, and "LotusScript" in Chapter 3.

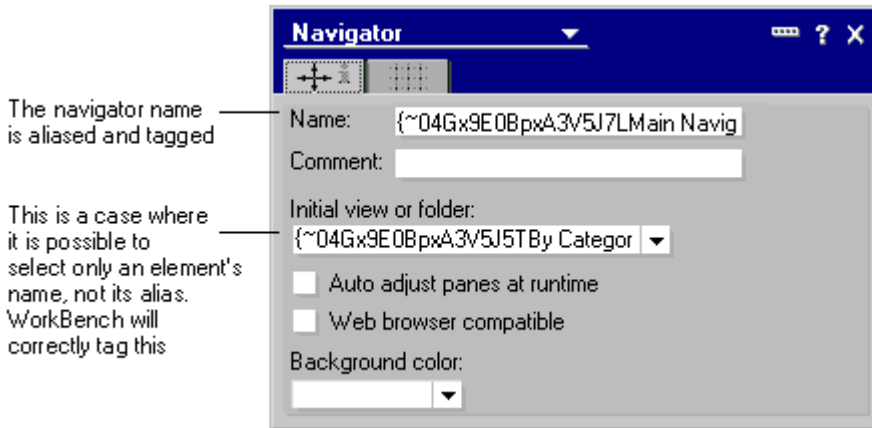
If you use standard alias prefixes, as recommended in "Using aliases" in Chapter 3, you can make the WorkBench ignore references to them in LotusScript automatically. However, there may be other references that must not be tagged, such as file names. Usually the simplest way to handle these is to let the WorkBench tag them, but mark them 'Prevent Translation' in the glossary.

Note During the building of language databases by the WorkBench, LotusScript is *only* recompiled if the design element that contains it has been selected for building and translation (marked ). In applications where script libraries are used it is important to consider that choosing not to build an element may cause problems because uncompiled scripts could contain translatable strings. This is particularly true if elements use a script library that contains string constants that are declared in the global declarations section. Also, remember that script libraries can use other script libraries so it is important to audit the script library usage structure before disabling elements from being built.

If you want to prevent the tagging of LotusScript, you can do so in the Tag the Following list box on the Tagging Options dialog.

Navigators

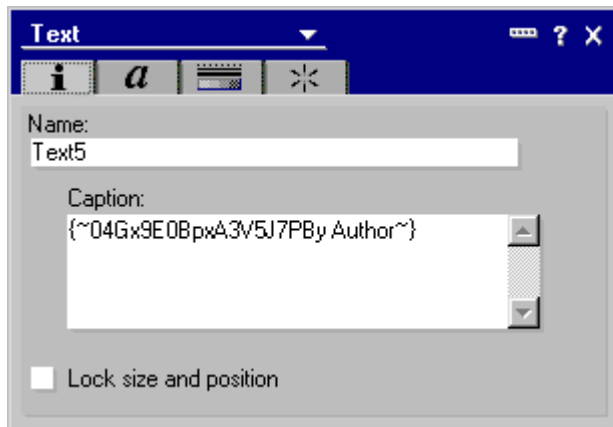
Navigator names can be visible to users, for example on the View - Show menu, so they should be tagged. They can also be used for computation, for example they can be referenced in other navigators, so they should be aliased. In the tagged database you will see something like this:



The initial view or folder is chosen from a drop-down which only lists element names, not aliases. The WorkBench tags the navigator name and (by using the Notes API) the entry in the 'Initial view or folder' drop-down.

Text boxes in navigators

Text in navigators is held in text boxes. It is visible to users and so should be tagged for translation. It is not used for computation and cannot be aliased. In the tagged database you will see something like this:



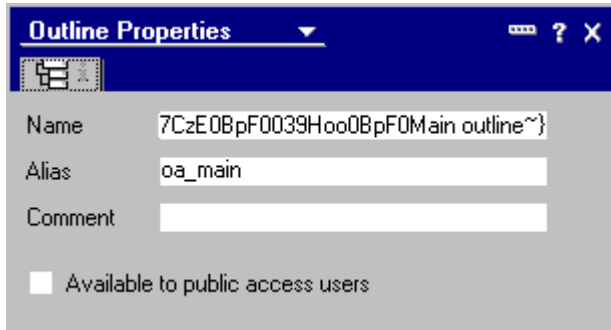
Web navigators

Any navigator that has its 'Web browser compatibility' option checked, will be ignored for building or tagging because Notes has to create the special Web-related \$NavImagemap field (the Web version of the Navigator). Domino Global WorkBench does not support this feature. Such navigators will be transferred unchanged to the language databases. If the Web option is unchecked a navigator will be treated normally and will build and be tagged as usual.

If you want to process a Web compatible navigator, you must uncheck the Web compatible option before creating the tagged database. Then, after building the language databases, reopen the Navigator using Domino Designer, and put back the Web compatible option in the language databases.

Outlines

Outline names can be visible to users so they should be tagged for translation. They can be used for computation, for example they can be referenced in formulas, so they should be aliased. By default the WorkBench does tag outline names. In the tagged database you will see something like this:



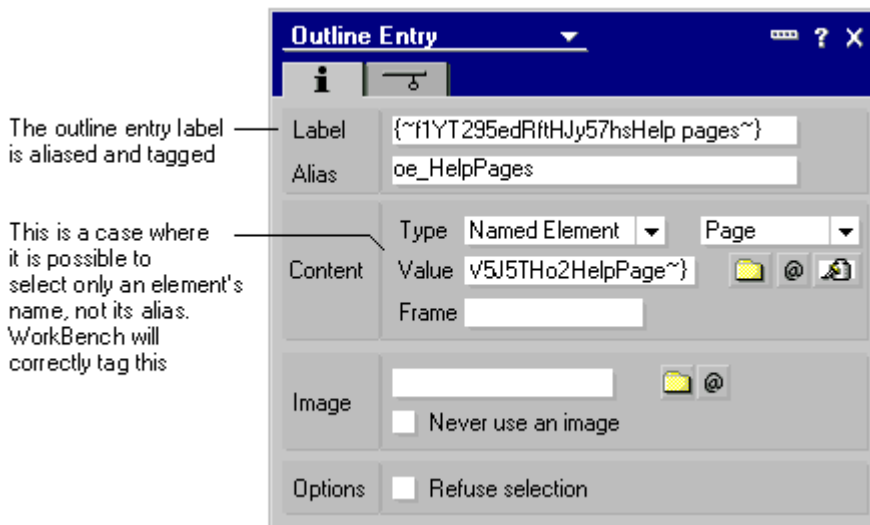
The screenshot shows the 'Outline Properties' dialog box. It has a title bar with a dropdown arrow, a menu icon, a question mark, and a close button. Below the title bar is a small icon of a document with a grid. The main area contains the following fields:

- Name: 7CzE0BpF0039Hoo0BpF0Main outline~}
- Alias: oa_main
- Comment: (empty text box)
- Available to public access users:

The 'Comment' text would usually be marked 'Prevent translation' in the glossary.

Outline entries

Outline entry labels can be visible to users so they should be tagged for translation. They can be used for computation, for example they can be referenced in formulas, so they should be aliased. By default the WorkBench does tag outline entry labels. In the tagged database you will see something like this:



The screenshot shows the 'Outline Entry' dialog box. It has a title bar with a dropdown arrow, a menu icon, a question mark, and a close button. Below the title bar are two icons: a person and a key. The main area contains the following fields:

- Label: {~f1YT295edRitHJy57hsHelp pages~}
- Alias: oe_HelpPages
- Type: Named Element (dropdown), Page (dropdown)
- Content: Value: v5J5THo2HelpPage~} (with folder, @, and image icons), Frame: (empty text box)
- Image: (empty text box), Never use an image:
- Options: Refuse selection:

Annotations on the left side:

- The outline entry label is aliased and tagged (points to the Label field)
- This is a case where it is possible to select only an element's name, not its alias. WorkBench will correctly tag this (points to the Value field)

The 'Value' is either a formula or the name of another design element, for example a page or a form. In the latter case, the design element is chosen from a drop-down which only lists element names, not aliases. The WorkBench tags the 'Value' entry by using the Notes API.

The WorkBench does not tag entries in 'Frame' or 'Image'.

Pages

Pages are used in frames or as banners. Their names do not usually appear to the end user, and therefore do not need to be translated.

Private design elements

Private design elements can only be seen by their author, and should not generally be tagged or translated.

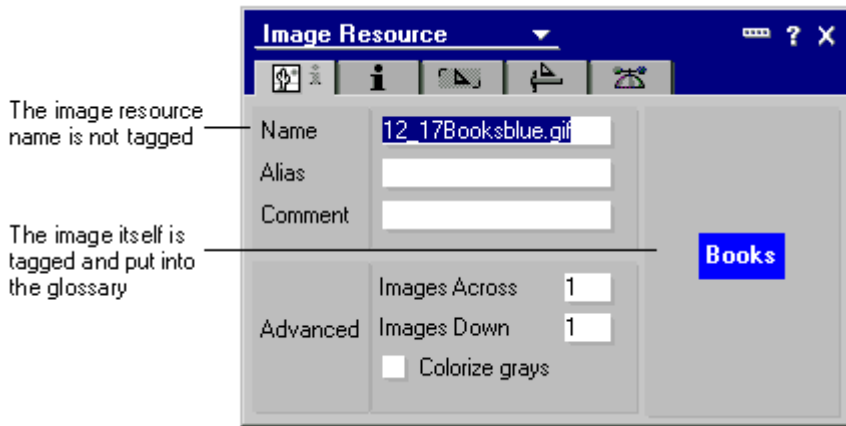
If you want to include private design elements in the tagging process you must select them on the Tagging panel of the Tagging Options dialog.

Resources

Notes Release 5 supports various resource types.

Images

The names of image resources (.GIF and .JPEG files) are not visible to users and therefore do not require translation. The WorkBench does not tag image resource names, or references to them inserted by Notes in forms. The image itself, however, is tagged internally by the WorkBench and is put into the glossary for possible translation.



See "Translating graphics" in Chapter 5 for more information.

Applets (Java resources)

Release 5.0.5 of the WorkBench processes .properties files in Java resources.

Shared fields

Shared fields should be handled in the same way as normal fields (above).

Script libraries

The WorkBench does not tag the names of script libraries. The library content will be tagged if it's LotusScript or Java.

About and Using documents

The WorkBench tags About and Using documents (Help Using) for translation. For additional information see "About and Using documents (Help About and Help Using)" in Chapter 3.

Actions

See "Actions", above.

Roles

Role names are not seen by database users and should not therefore be tagged or translated. The WorkBench does not tag role names.

'Shared / Personal on first use' folder or view

Any private folders or views that have been created by opening 'shared / personal on first use' folders or views should be removed from the source database before you create the tagged database because they will not be accessible in the built language databases. Users of the language databases will create their own private folders or views by opening the localized version of the 'shared / personal on first use' folder or view.

Purely private design elements are never processed by the WorkBench's tagging or build functions.

When a user opens a 'shared/personal on first use' folder or view for the first time, Notes creates a private folder or view with the same name (which the user can customize) and hides the public one. If the name of the public folder or view changes for any reason, for example by translation, Notes will no longer be able to locate the original private version and will display the public one again. This will create a new private copy one when opened.

To go back to his or her original customized folder or view, a user must:

- Make a note of the name of the new private folder or view created by Notes.
- Delete it.
- Rename the original folder or view with the name noted above.

Special folder and form names

The following folder names, and references to them, should not be translated: (\$Alarms), (\$All), (\$Calendar), (\$Drafts), (\$Inbox), (\$Meetings), (\$Profiles), (\$Sent), (\$ToDo), and (\$Trash). However, the WorkBench will tag them so you should take care to mark them Prevent Translation in the glossary.

Similarly, you should not translate special form names that begin "\$\$", for example \$\$ViewTemplateDefault and \$\$ViewTemplate.

Static text

Static text is visible to users and should be tagged. It is not used for computation, and it cannot be aliased.

If you use 'plain text' glossary entries for text that contains attribute changes, you may have to restore some of the attribute changes in the language databases after building. When using plain text glossary entries, the WorkBench preserves attribute changes in static text by tagging it in separate sections. See "Handling attributed text (rich text)", above.

Subforms

Subform names are never displayed on the screen, so they should not be tagged. The WorkBench does *not* tag subform names. However, text within subforms should generally be tagged and translated.

Caution If a subform has been embedded in a form (rather than computed), *and you are working on a Notes 4.x database or template*, take care not to translate its text within the host form. At run time Notes always refreshes the subform part of the form by referencing the original subform, so any subform translations entered in the host form will not be seen. WorkBench *does* tag this text. You should mark it as 'Prevent Translation' in the glossary. A better solution is to compute the subforms rather than embed them. The WorkBench recognizes embedded subforms in Notes Release 5 databases or templates and does not tag them when they appear in a host form.

See "Subforms" in Chapter 3.

URLs

The WorkBench will tag URLs only when they appear within quotes in formulas. In general they should not be translated, but sometimes they do contain displayed text. For example, in the following, you should translate the underlined text (it would not be underlined in a real URL):

```
<a href=/dbname/($All)?OpenView>All Documents</a>
```

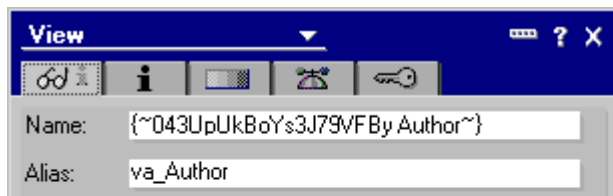
```
<a href=/dbname/Archiving?OpenView>Archiving</a>
```

```
<a href=/dbname/by+Author?OpenView>by Author</a>
```

Note There are two types of text here: firstly, the function action specifying the view to open (the part before the ?); and secondly, the name displayed by the browser (the underlined part). Only the displayed name should be translated. Also notice that in the first part, spaces are replaced with the + sign, because the Internet does not support empty spaces.

Views

View names can be visible to users, for example on the View menu, so they should be tagged. They can also be used for computation, for example they can be referenced in formulas, so they should be aliased. In the tagged database you will see something like this:



When tagging views within views, the name of the view and its parent view must be tagged separately to maintain functionality.

Columns in views

Column names are visible to users, so they should be tagged. They are not used for computation, so they do not need to be aliased.

Cascaded views

Cascaded views (such as Views\View1 and Views\View2) may not display correctly in a tagged database. This is because the WorkBench tags them as separate pieces of text. If you have selected 'Create new terms - Always', the cascaded part will be tagged with two different IDs and will therefore not appear under the same category when the tagged database is opened in Notes.

Tagging limitations

Notes has certain limitations on the length of particular pieces of text (see "Length limitations" in Chapter 3). This means that:

- Because a Domino Global WorkBench tag adds 21 characters to the length of a piece of text, the labels in a tagged database may appear truncated. However, in the language databases you build from it there will be no truncation.
- If the translated version of a piece of text exceeds the allowed length, there will be a truncation in the built database (in that language).

Items the WorkBench does not tag

The WorkBench does not tag:

- Aliases.
- Field names.
- Subform names.
- Navigators where the Web option has been selected.
- Role names.
- View selection formulas.
- Password fields.
- REM statements

And, unless you select them on the Tagging panel of the Tagging Options dialog box, WorkBench does not tag:

- Database titles.
- Template names.
- Database class names.
- Hidden design elements.
- Private design elements, such as views and folders (these are only seen by the people who created them).

Chapter 7

Checking and using built databases

This chapter explains how to check the functionality of built language databases and make any necessary adjustments to formatting and layout. Also described is the procedure for switching languages in a multilingual database.

Note For translation workflow processes, see "Setting up translation workflow" and "Using translation workflow" in Chapter 8.

Functionality in language databases

There are various things, such as format and layout, that you will probably have to adjust by hand in the built language databases. However, before you do this you should make sure that the language databases function correctly. A language database will not function properly if any of the links between its design elements has been broken. This typically happens because:

1. A reference in a formula or LotusScript to an alias has been translated, but the alias itself has (correctly) not been translated.

Solution Mark the corresponding glossary entry 'Prevent Translation'. Language databases should then function correctly.

2. A reference in a formula or LotusScript to a design element name (not an alias) has been translated, but the name itself has not been translated.

Solution Either tag the design element name for translation, or, if it must not be translated, mark the reference in the formula or LotusScript as 'Prevent Translation'. Provided both pieces of text are translated the same way, the language databases will function correctly.

3. A reference in a formula or LotusScript to a design element name has not been translated, whereas the name itself has been translated.

Solution Create an alias for the design element that is identical to the untranslated reference in the formula or LotusScript. Provided both pieces of text, the reference and the name itself, are translated the same way, the language databases should function correctly.

4. A reference in a formula or LotusScript to a design element name has been translated, and so has the name itself, but the translations are different.

Solution The language databases will only work correctly if the translations are the same. Rebuild the tagged database, first setting the prompting options to 'Prompt when - More than one term is found'. Make sure you tag both occurrences of the text to the same translation.

After correcting any of these problems you will need to rebuild or update the language databases. See "Updating" and "Rebuilding databases" in Chapter 6.

A language database built in the reference language should be identical to the original database. To check on the effects of translation before you start the actual translation work you can create a pseudo-translated language database (see Chapter 4).

Design changes in language databases

The following features of the WorkBench help to preserve the layout of built databases when they are updated:

- A 'Preserve layout' option on the Build Options dialog box makes the WorkBench save as much layout information as possible before updating a language database, and then reapply it once the updating operation is complete. Selecting 'Preserve layout' therefore eliminates much of the need for reworking of layout when databases are updated.
- The update function operates at the field level, so as soon as a particular part of the database design is final you can safely carry out the post-build work for it in the language databases. If a design element (for example, form or view) does subsequently change, and 'Preserve layout' is unchecked, you may have to repeat the work, but only on the fields that have changed.

The layout and static text of a form is held in a single field called \$Body. So if a piece of static text in a form changes, and 'Preserve layout' is unchecked, the layout of the form will need to be reworked on update.

Whenever you make changes in a language database, consider whether a change should be made in the source database or the glossary to prevent the problem in the future. For example, if you modify a translation for reasons of space, you may want to put the shortened version of the text into the glossary.

The WorkBench only updates databases if *design elements* have changed. However, if you make changes to glossaries only, you can force an update by editing the source database. See "Forcing an update" in Chapter 6.

Checking built language databases

The following sections describe some of the things you should look out for in built language databases.

Action bars

Check that all the buttons display properly.

Action names

Check that they are all translated.

Attributes

You should check the text attributes (bold, italic, color, fonts and so on) in the language database. Look for the following:

- Attributes that have been incorrectly transferred from the source database. This can happen if you use plain text rather than rich text glossary entries (as in Notes Global Designer Release 4.6) for attributed text. See "Handling attributed text (rich text)" in Chapter 6.
- Attributes that have been correctly transferred from the source database but are not suitable in the language database. For example, it may not be appropriate for a word that was bold in the source database to be bold in all of the language databases (for reasons of space, local style, and so on). These may therefore need to be changed in the language database.

You can change fonts during building by using the 'Replace fonts' option in the build dialog.

Buttons

Check that buttons align properly.

Framesets

Check that the text fits properly in each of the frames.

Graphics

Check for truncations.

HTML

If your source database includes HTML that is not supported by Domino Global WorkBench, you may have to translate it separately in each language database. See "HTML" in Chapter 3.

Layout

If your source database is in English it is likely that the text in your translated language databases will be longer than in the original. You will therefore probably have to make some layout adjustments to accommodate the longer pieces of text. And you may have to move text and fields, and adjust column widths, to suit the language.

If potential expansion was taken into account when the database was designed this task will be diminished.

Launch options

Go to the Launch tab in Database Properties. Under 'When opened in the Notes Client' and 'When Opened in a Browser' and select the *translated* version of the proper Frameset or Navigator.

On-disk structure (ODS)

The ODS is the format in which a Notes database is physically stored on disk. It is sometimes called the 'database format'. In Release 5, the ODS version of a database is listed on the Info tab of the database properties box.

You must make certain that your built databases have the same ODS version as their parent source database. The WorkBench, by default, builds Release 5 databases. If you encounter discrepancies here it usually means that you have used the WorkBench to build from a Release 4.6 source without following the necessary setup procedure.

See "Using Domino Global WorkBench with Notes Release 4.6 databases" in Chapter 1.

Outlines

Check font sizes and spacing, and look out for truncations.

Private views

Usually you should delete any private views from the built language databases.

Punctuation derived from formulas

When they are used to construct phrases or sentences, formulas sometimes contain strings composed of punctuation marks only. These are not tagged by the WorkBench, but they may need to be localized. For example, in French a colon is normally preceded by a space.

If you encounter any problems of this sort, you will have to 'translate' the punctuation string in the built language database(s).

Tables

Check that there are no bad text wrappings or misalignments. Check that tab characters are appropriate for the translated text.

Truncations

You will see length warnings if the build process has to truncate any text it puts into a language database (see "Length limitations" in Chapter 3). You should shorten any translations that lead to these warnings and then rebuild.

View columns

Check that the columns are wide enough to display their title and contents.

Switching languages in built databases

The method of switching languages in a Domino Global WorkBench database depends on the type of database it is:

1. **Unilingual databases.** These contain design elements in one language, like any standard Domino database. If you are using a set of unilingual databases to provide similar information in several languages (for example a Web site), doc links or hotspots should have been included in the source database design to lead from one database (language) to another. This is what the Language SwitchBar does for synchronized databases. To switch languages, click the appropriate link, or section of the SwitchBar, to go to the database for the language you want to see.
2. **Multilingual databases.** These contain design elements in more than one language. Users see the *design elements* in their preferred language (or the default language if their preferred language is not supported by the database).

The forms and views in a multilingual database are often designed so that *data documents* in the database also display according to language. So a user whose preferred language is German sees only German documents. The language of a document is specified in a field called \$Language, and it is typically the language of the form that was used to create it. Synchronized databases handle the \$Language field automatically.

The next section explains how to change your preferred language.

See "Multilingual databases" and "Synchronized databases" in Chapter 1, and "Language synchronization" in Chapter 8.

Switching to a different preferred language in a multilingual database

The procedure for switching languages in a multilingual database depends on how you are accessing the database.

If you are accessing the database using Notes

The procedure is:

1. Choose File - Preferences - Notes Preferences.
2. Tab to the International panel. You will see three separate language settings:
 - 'Regional Settings'. This is the language that Notes uses to set defaults for such things as collation, units of measurement, dates and so on. See Notes Help if you need more information.
 - 'Alternate Name Language'. This affects the way usernames are displayed. See Notes Help if you need more information.
 - 'Content Language'. This is the setting that Domino Global WorkBench uses to determine your preferred language. Its default value is the 'Regional Settings' language.

3. Select your preferred language under 'Content Language'. If you select more than one language, organize them in order of preference, with your most preferred language at the top of the list.
4. Click OK.
5. Click OK again.

If you are accessing the database using Netscape Communicator® or Netscape Navigator®

The procedure is:

1. Choose Edit - Preferences.
2. Select Navigator.
3. Select Language.
4. Add the language(s) to the list.
5. Click OK.

If you are accessing the database using Internet Explorer®:

The procedure is:

1. Choose View - Internet Options - General.
2. Click Languages.
3. Select one or more languages. Arrange them on the list (by using the Move Up, Move Down and Add buttons) so that your first choice is top of the list, your second choice is second and so on.
4. Click OK.
5. Click OK again.

You will see documents in the highest selected language that is available in the database. If none of the languages you selected is supported by the database, you will see documents in the default language.

Note You must reload the current page to see the change.

Chapter 8

Language synchronization and translation workflow

In a multi-language database environment, such as an international Web site, translation of documents is usually a requirement. New documents authored in one language must first be copied and then translated into the other Web site languages. Domino Global WorkBench can help you to automate much of this process.

'Language synchronization' automates the copying part of the process. When you add language synchronization features to your built databases, and run the Language Synchronizer (a server task), translatable documents authored in one language are automatically copied to other languages. For example, if you have a set of unilingual databases supporting US English, German and Japanese, a document created in the US English database will be copied by the Language Synchronizer to the German and Japanese databases. The process is essentially the same for a multilingual database, only there the different language versions of the document are held in the one database.

Once copies of a newly authored document have been created for different languages, those copies will have to be translated. To streamline translation, you may want to adopt a 'translation workflow' process. There are two ways of doing this:

1. You can set up your own workflow process. When it creates a synchronized database, the WorkBench adds a hidden subform to each form in the database. The fields created by the subform in any new document are used by the Language Synchronizer to identify new translatable documents and to copy them ready for translation. You can make use of the subform fields to set up a workflow application and to select documents for display.
2. You can use Domino Global WorkBench's translation workflow. To do this you first build your databases as synchronized databases and then create a translation workflow database, based on a template supplied with Domino Global WorkBench. The translation workflow database controls the sequence of translation activities. A person is nominated as translation manager for each language pair and, with translation workflow operating, the relevant manager is automatically mailed whenever any new translatable document is created in any language in your Web site. The manager assigns a translator who will translate the new document, one or more reviewer(s) who will review the translated version, and an approver who will give final approval. Each person is mailed automatically when the previous step in the workflow has been completed. Only when a translation has been approved will it become visible to users of your Web site.

If your translators are globally scattered you may prefer to build unilingual synchronized databases rather than a single multilingual one. This is because, with Release 5.0.5 of Domino Global WorkBench, unilingual synchronized databases can be placed on *any* server; they are no longer restricted to a single directory on one server. The databases can therefore be put on servers in the most geographically suitable locations.

Note Translation workflow is concerned only with translatable documents in the working databases built by Domino Global WorkBench. It is *not* concerned with the localization of database design which is handled by the regular Domino Global WorkBench tagging and glossary processes.

Language synchronization

If you have used the WorkBench to build a set of language databases (or a multilingual one) you can add language synchronization features to the database(s). The Language Synchronizer, a Domino Release 5 server task, will then synchronize documents across languages in the database(s). This means that every time a document is created in one language, a copy of it will be created for each of the other languages, provided the original document was flagged as 'translatable' or 'global' (see "Synchronization states" below). Documents created by the Language Synchronizer from translatable documents will need to be translated.

If the original version of a document changes, the Language Synchronizer makes appropriate changes to the corresponding copies in the other language databases. And if the original version is deleted, the Language Synchronizer deletes the corresponding copies in the other language databases.

You can add language synchronization features to any multilingual language database or to a group of unilingual ones, provided they are databases that have been built by the WorkBench from a single source database. Language synchronization can only take place between unilingual databases or within a multilingual database; a unilingual database cannot be synchronized with a multilingual one.

The Language Synchronizer is scheduled from a central database, the Synchronizer configuration database, which contains information about all your synchronization sets (a 'synchronization set' is a group of unilingual databases, or a single multilingual one, in which documents are being, or can be, synchronized between languages). The Synchronizer configuration database allows you to add or remove databases from synchronization sets, enable or disable language synchronization, or move synchronized databases from one server to another or from one directory to another. It also allows you to schedule when and how often language synchronization should happen, and provides a log and some useful statistical information about each synchronization set.

You add language synchronization features to databases using the WorkBench, which does the following:

- To the end of every form in each database being synchronized it appends a hidden subform. This contains various language-related fields, including \$Lng_SynchType, \$Lng_State and \$Language. For a list of these see "Fields in synchronized databases", below.
- To each database being synchronized it adds agents that allow users to change the synchronization state of documents (see below).
- It adds information about the other databases being synchronized (if any). This information is used by the setup procedure in the Synchronizer configuration database.

Synchronization states

The subforms added to the forms in each database allow every document created in the synchronized databases to have one of four states (\$Lng_SynchType). When you create the synchronized databases you select a default state for documents created from each form. The four states are:

1. Local. This indicates that documents created from the form are to be restricted to the languages in which they are created. They are never copied to other languages.

Note Local is the default state. Users can change the state of documents after creation, and it is therefore usually safest to choose Local to start with: once a document has been created, edited and checked its state can be changed to Translatable or Global if necessary.

2. Translatable. This indicates that documents created from the form are required in the companion languages, and that they will need translation. When the Language Synchronizer runs, it copies newly created translatable documents to the other languages and marks them Untranslated (\$Lng_State=0).

If the Language Synchronizer encounters an existing translatable document that has changed it marks the versions in the other languages as Modified (\$Lng_State=3) but otherwise does not change them. Only changes to the document that was originally created can trigger changes in other languages.

In multilingual databases, new documents are created in the same database; in unilingual databases, new documents are created in the other language databases. The documents have a field, \$Language, that contains a language identifier.

3. Global. This indicates that documents created from the form are required in the companion languages, but that they do not need translation. When the Language Synchronizer runs, it copies newly created global documents to the other language databases. If the original version of a global document changes, the Language Synchronizer will automatically update the copies in the other language databases.

In a multilingual database there is only one copy of each global document. The \$Language field is set to include all the languages supported by the database.

4. No selection. This indicates that you do not want the Language Synchronizer ever to process documents created from the form.

Note Such documents are handled like local document, but they cannot be made global or translatable at a future date.

The Language SwitchBar

If you create synchronized unilingual databases, you further have the option of including a 'Language SwitchBar' which provides links to other language versions of documents created from the form. Users simply click the language they want to see on the SwitchBar.

The SwitchBar is available only in synchronized *unilingual* databases. In multilingual databases, the form used to display a document is based on the user's language, so it makes little sense to move to another language version of a document when the form used to display it will not change.

Note The Language SwitchBar includes links to *all* language versions of a document, regardless of whether they have actually been translated.

See "Displaying the Language SwitchBar", below, for more information about SwitchBars.

Synchronization reports

After language synchronization a log is written to the Synchronizer configuration database, which can be viewed either on its own or through the History tab of the relevant synchronization set document. The statistics appropriate to the synchronization set are also updated and can be viewed in the same database.

Implementing language synchronization

The basic steps are:

1. Prepare the source database. The source database will need to contain the appropriate forms and views for your application. When you add language synchronization features to a language database built from the source database, a hidden subform is appended to each of its forms. The hidden subform contains fields that you will need to reference in the original source database design, for example in view selections.
2. Install the Language Synchronizer task and the Synchronizer configuration database on a server. This is a job for the server administrator. The Synchronizer configuration database is installed in the server's Domino data directory.
3. After building language database(s), add the language synchronization features to them. You use the WorkBench to do this.
4. Deploy the synchronized database(s) as a 'synchronization set' on one or more servers, using the Synchronizer configuration database to supply details of the synchronization set and the desired language synchronization schedule to the Language Synchronizer server task. You can use the WorkBench to make the initial entries in the Synchronizer configuration database.

Note A 'synchronization set' can be a single multilingual database or a number of unilingual databases, each supporting a different language.

Preparing a database for language synchronization

The following paragraphs contain advice for developers of source databases that will be built as synchronized databases. This is in addition to the normal considerations when preparing databases for processing with the WorkBench.

If you intend to use language synchronization as a basis for your own workflow application, you will find it helpful to refer to the sections on "Translation workflow", below, which explain how the Domino Global WorkBench workflow application functions. If you plan to use the Domino Global WorkBench translation workflow you should look at the section "Preparing a database for translation workflow". For more general information about using built synchronized databases, see "Using synchronized databases" in this section.

Views for translators

You are likely to need an 'untranslated' view for translators. This should be a view similar to one of your main views but including a selection formula like this:

```
SELECT $Lng_State = "0"
```

If you want a view for documents that were translated but for which the original has been modified (you could alternatively include these documents in the previous view) you need a selection formula like this:

```
SELECT $Lng_State = "3"
```

This view might also contain action buttons for changing the state of documents, in particular to change \$Lng_State to "1" (translated) or "0" (untranslated).

Forms for translators

Depending on the content of the documents being translated, and the complexity of the original form used, you may wish to provide a special form for translators. A form which, for example, makes editable only those fields which are known to require translation. Alternatively you may wish to use the original form but restrict editing by translators in this form.

Any translator-specific form might also contain action buttons for changing the state of documents, in particular to change \$Lng_State to "1" (translated) or "0" (untranslated).

Selecting translated documents in views

It is probable that you will not want users (except for translators) to see documents that have not been translated. It may therefore be advisable to modify view selection formulas to include:

```
$Lng_State = "1" (see "Fields in synchronized databases", below, for details about $Lng_State).
```

Note You may also want to select documents with \$Lng_State of "5". These are the documents that do not require translation (local and global documents). And you may want to select documents that have been modified and/or marked for deletion (\$Lng_State "3" and/or "4").

In order to display documents in the source database (for testing prior to building) you may want to modify this to:

```
$Lng_State = "1" | @IsUnavailable($Lng_State)
```

View selection formulas in multilingual databases

Because multilingual databases contain documents in many languages, if you want just one language to appear in a given view (as you probably will) it is a good idea to add language selection to the view selection formula. For example:

```
SELECT $Language = "en-US"
```

See "Multilingual databases", in Chapter 1.

Displaying the Language SwitchBar

If you choose to include a SwitchBar when you add language synchronization features, the WorkBench adds a frameset to each of the language databases. The frameset has two frames named Main and Switchbar. 'Translatable', 'global' and 'local' documents are displayed using this frameset: the document content appears in the Main frame and, if the document is translatable, the SwitchBar appears in the Switchbar frame.

Domino Global WorkBench does *not*, however, overwrite any existing frameset setting. So if any form for a translatable, global or local document in your source database is already set to launch a frameset you will need to make sure that the frameset has a frame for content named Main and a frame for the SwitchBar named Switchbar. This provides you with a way of customizing the positioning of the SwitchBar (which normally appears at the bottom of the display).

The SwitchBar is made up of clickable elements, one for each language in the synchronization set. There are three types of clickable element:

1. Rich text. This is the default: the name of the language.
2. Graphic. The country flag.
3. User defined.

These are specified in the database DGWSTORE.NSF, in the Domino Global WorkBench program directory. To modify them, open the appropriate Language Definition document(s) in that database and tab to Available Formats. Add carriage returns (CRLF) to the clickable elements if you want to create a vertical SwitchBar.

A SwitchBar is built dynamically from elements of one of these three types (not a mixture). You specify whether you want the rich text, graphic or user defined version by clicking the appropriate button in the Language IDs view of DGWSTORE.NSF.

Note In order for the SwitchBar to be displayed on Web browsers, the following properties must be checked:

- On the first tab of the Database infobox, 'Web access: Use JavaScript when generating pages'.
- For each form, on the second tab of the Form properties infobox, 'Generate HTML for all fields'.

Selecting document type on compose

When you add language synchronization features using the WorkBench, you choose whether or not authors are to be allowed to specify the initial type of the documents they create: local, global or translatable. You make this choice separately for authors using Web browsers and for authors using the Notes client. Authors choose a document type either from a dialog box (Notes client) or by selecting a radio button at the bottom of the document (Web browser).

After creation, agents ToLocal, ToGlobal and ToTranslatable in the language database allow Notes client users to change the type of their documents.

If you want to enable type selection and you want it to be more of an integral part of the form you can instead add a field to the form definition which sets up the \$Lng_SynchType field as appropriate.

Caution Do not create a \$Lng_SynchType field in your form: it is present in the subform that the WorkBench adds to each form in a synchronized database.

Changing the type of a document after creation

In the Notes client, agents ToLocal, ToGlobal and ToTranslatable, which the WorkBench adds to synchronized language databases, allow users to change the type of their documents.

If you want to allow similar type switching from Web browsers you will have to add functionality to the relevant forms to change the value of the \$Lng_SynchType field as necessary.

Fields in synchronized databases

Fields used by the Language Synchronizer are held in the synchronized documents themselves, and in a profile document called "\$Lng" that WorkBench inserts into every synchronized database. The Language Synchronizer automatically makes appropriate entries in these fields.

Fields in synchronized documents

Synchronized documents contain the following fields that are used by the Language Synchronizer. They are contained in a hidden subform which the WorkBench adds to each (selected) form in the language database when it adds language synchronization features.

Field	Description	Values
\$Language	The language of the document.	For example, en-US. See "Adding new languages" for a list of identifiers. Note that, in this field, you must use the lowercase, hyphen, uppercase format.
\$Lng_UNID	An ID that is shared by all other versions of the document.	
\$Lng_SynchType	The synchronization state of the document.	"0" - Local "1" - Global "2" - Translatable
\$Lng_Original	Whether or not this is the originally created version of the document (and not just a translation).	"0" - Not the original "1" - The original
\$Lng_State	The translation state of the document. Translation workflow uses this field. Note All Local and Global documents have \$Lng_State=5.	"0" - Untranslated "1" - Translated "2" - Workflow in progress and this is the translator's working copy (the document is being translated, but translation is not complete) "3" - Modified (the document has been translated, but the original has now been modified) "4" - Deleted (the Language Synchronizer will first delete all language versions and then delete this document) "5" - The document is not a translatable document "6" - Archived (not yet implemented).
\$Lng_LastSynchronised	The last time the document was synchronized. Used for optimization.	
\$Lng_LastModified	Used for optimization.	
\$Lng_WorkflowState	Reserved for future use by translation workflow.	
\$Lng_OriginalLang	Language of original (authored) document.	
\$Lng_CopyUNID	Used internally to track synchronized copies and prevent users from creating copies not tracked by the synchronizer.	

Fields in the profile document

The profile document contains the following fields:

Field	Description	Values
\$Language	The language of the database (a list of languages in the case of multilingual databases).	
\$Lng_Databases	A list of the databases being synchronized.	
\$Lng_Id	The IDs of the languages being synchronized.	For example, en-US, de-DE, fr-FR.
\$Lng_Names	The full names of the languages being synchronized.	For example, English (United States), German (Germany), French (France).

Field	Description	Values
\$Lng_Switchbar	Whether or not the Language SwitchBar is being used.	"0" - No "1" - Yes
Database_Server_XXXX	Server of related language database.	
Database_File_XXXX	File path of related language database.	
Database_URL_XXXX	URL of related language database.	

Installing the Language Synchronizer and the Synchronizer configuration database

The Language Synchronizer task runs on a Domino server, and carries out the synchronization of documents between databases that contain language synchronization features. The following paragraphs explain how to install it. As part of the process, the Synchronizer configuration database is installed in the server's Domino data directory.

Note The Language Synchronizer must be installed on a Domino Release 5 server running Windows NT Release 4.0 or later.

- To install the Language Synchronizer run Setup from the 'Apps\DGW\Sync install' subdirectory on the Notes/Domino CD. You can choose to have install add the Language Synchronizer to the list of tasks for the server to launch automatically at startup.
- To start the Language Synchronizer, enter the following at the server console:

```
LOAD nsync
```

See "Running server tasks" in the Domino 5 Administration Help for more general information.

Once the task is running, the scheduling of language synchronization is controlled by synchronization set documents in the Synchronizer configuration database.

- If you need to stop the server task, enter the following at the server console:

```
TELL nsync QUIT
```

Adding language synchronization features

You can use the following procedure to create synchronized databases for the first time, to add to an existing group of synchronized databases, or to update synchronizing features in databases.

Note For information about synchronizing templates rather than databases, see "Synchronizing templates", below.

The steps are:


1. Prepare your source database for language synchronization (see "Preparing a database for language synchronization", above).

Note Language synchronization features can only be applied to databases built by Domino Global WorkBench, and they must be built from the same source database.

2. After creating (or updating) your tagged database, build the language database(s) that you want to synchronize in the usual way.

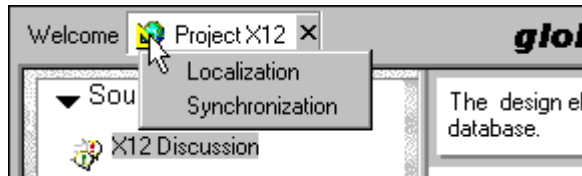
Note The Language Synchronizer cannot distinguish between new source documents and copies of them created by replication to other databases. Synchronized databases should not therefore be allowed to replicate together. On the Create Language Database dialog box choose 'Create database copy' as the Database Type.

3. Do one of the following:

- Choose View - Switch to Synchronization,
- Click  at the top right of the screen

or

- Click the WorkBench icon of the project whose databases you want to synchronize and select Synchronization:



The Language synchronization display appears.

4. Select the source database that corresponds to the language database(s) you are going to synchronize. The language databases are displayed.
5. Click to the left of a language database icon to select it for language synchronization. A tick indicates that the forms in a database will be synchronized, subject to the selections you make below. If your databases are unilingual you will need to select more than one of them to form a 'synchronization set'. If your database is multilingual, select it alone.

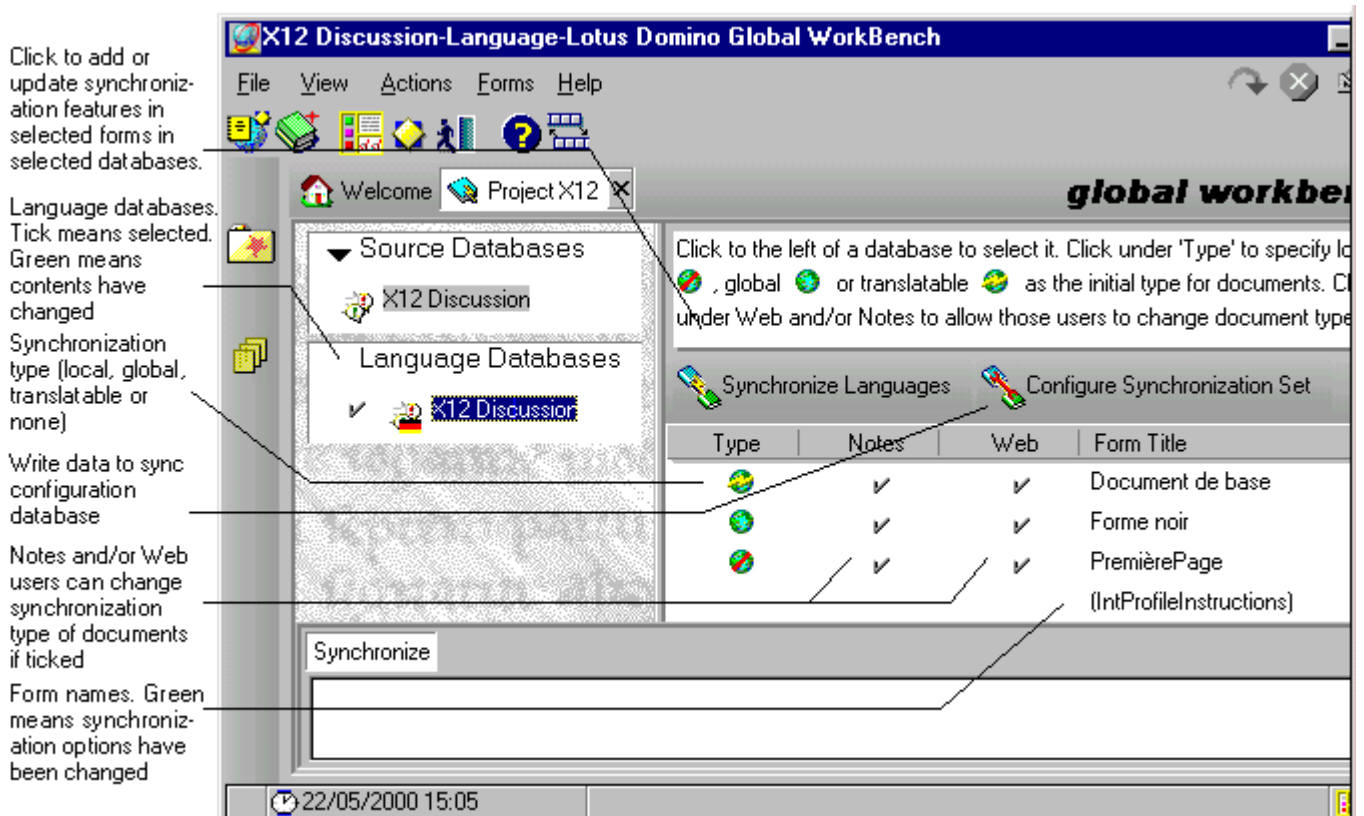
You cannot synchronize unilingual databases with multilingual databases or multilingual databases with other multilingual databases. If you want to have multiple synchronization sets, create separate projects with the same source database. This can be done by creating a new project, adding the source and then adding existing tagged and language databases.

A language database that has been rebuilt or updated by the WorkBench after language synchronization features have been added to it, is displayed in green. The process described here will update its language synchronization features.




Caution If a database has changed, look out for any new forms that may have been added to it: they will be displayed in black, not green.

6. To the right of the display is a pane that displays the forms in the database. Make selections here to determine the default synchronization type of documents created from the forms and to specify whether or not you want authors to select a synchronization type when they compose new documents from the forms.

Note You need to make selections for *one database or language only*. The selections made for a form will be used for that form in all languages. If you right-click in the 'Form Title' column, you will see a pop-up menu that allows you to mark *all* forms in the database in one of these ways. You can also set the language synchronization options of a form by tabbing to 'Synchronizer' on its properties box.



Click in the 'Type' column to set the initial synchronization status for documents created from the form (repeated clicking will cycle through the options). The options are:

- 'Local' if documents created from the form should not be viewed in other languages: .
- If you allow users to change the synchronization type of documents (see below), local is a good setting to choose here. Users can then keep their own documents local while they write and edit them. Only when the document type is changed to global or translatable will the Language Synchronizer make it available to users other languages.
- 'Global' if documents created from the form should be passed to other languages, but do *not* need translation: .
- 'Translatable' if documents created from the form should be passed to other languages, and *do* need translation: .
- No selection if you do not want the synchronizer ever to process documents created from the form. The WorkBench will not add the synchronizer subform to these forms, and it will therefore not be possible to change the status of documents created from them to Local, Global or Translatable (see Using synchronized databases).

Click in the 'Notes' and/or 'Web' columns to add or remove ticks:

- A tick in the 'Notes' column means that users working from a Notes client will be able to set the initial synchronization type of any document they create using this form. When they save the document a dialog box will give them the option to make the document local, global or translatable.

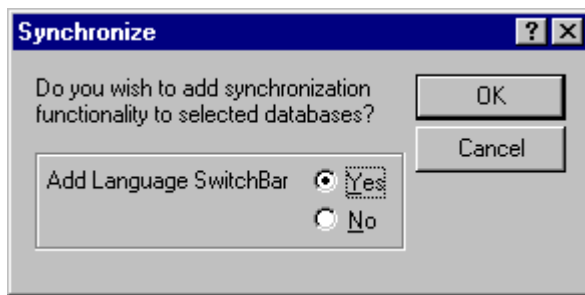
- A tick in the 'Web' column means that users working from a Web browser will be able to set the initial synchronization type of any document they create using this form. Radio buttons at the end of the document will give them the option to make the document local, global or translatable.

If, after synchronizing, you make a change to any of the settings in the 'Type', 'Notes' or 'Web' columns, the form's name will be displayed in green to indicate that its language synchronization features need updating (click Synchronize Languages or choose Actions - Synchronize Language Databases).

Forms that have not been synchronized before, or whose language synchronization settings have not changed since the last time they were synchronized, are displayed in black.

Right-clicking in the 'Form Title' column displays the Forms menu as a pop-up.

7. Click Synchronize Languages or choose Actions - Synchronize Language Databases. The following dialog appears:



If you are synchronizing unilingual databases and you want to include a Language SwitchBar on translatable documents, select Yes. (The SwitchBar option is not available when building multilingual databases.)

8. Click OK. The language synchronization features are added to (or updated in) all the ticked language databases, subject to the selections in the 'Type', 'Notes' and 'Web' columns.

Note As part of this process, appropriate translated versions of the ToLocal, ToGlobal and ToTranslatable agents are copied from DGWSTORE.NSF to the language database(s). If a translated version is not available you will see a warning to the effect that the default language version is being used instead. You can create language versions by opening DGWSTORE.NSF in Domino Designer, making a copy of an existing agent, translating its title, and then changing the contents of the \$Language field to the appropriate language identifier, for example jp-JP (see "Adding new languages" in Chapter 4 for a list of language identifiers).

In an update scenario, if you have deselected a form that was previously synchronized (in other words it was local, global or translatable), the WorkBench will remove the language synchronization subform from it.

9. (Optional) Click Configure Synchronization Set and complete the Configure Synchronization Set dialog box. This puts the necessary information about the synchronization set, and the required language synchronization schedule, into the Synchronizer configuration database.

Alternatively you can configure the synchronization set manually by using the Synchronizer configuration database directly.

Note For configuration to be successful, you must have manager access to each of the language databases and editor access to the Synchronizer configuration database.

The screenshot shows a dialog box titled "Configure Synchronization Set" with the following fields and annotations:

- Server Name:** A dropdown menu containing "BCA_GENL/BCA/A/XYZCorp". An annotation points to it: "The server on which the Synchronizer task runs".
- Name:** A text field containing "X12 Discussion". An annotation points to it: "A name for the synchronization set".
- Description:** An empty text field. An annotation points to it: "A description for the synchronization set".
- Synchronization Schedule:** A group box containing:
 - Repeat Every:** A text field with "1" and a dropdown menu with "Day(s)". An annotation points to it: "The schedule for the Synchronizer server task".
 - at:** A text field with "12:01 AM".
- Enable Synchronization for this Synchronization Set:** A checked checkbox. An annotation points to it: "Start synchronizing as soon as language databases are deployed?".

Buttons for "Set", "Cancel", and "OK" are visible on the right side of the dialog.

10. If you want to return to the localization display, choose View - Switch to Localization.
11. Install the synchronized database(s) on the appropriate server(s).

Synchronizing templates

The synchronization features of Domino Global WorkBench are designed to be used with databases rather than templates. Because the WorkBench puts important information in profile documents, and these are not passed on from templates to databases, you cannot use databases created from synchronized templates with the Synchronizer server task. If you wish to synchronize databases created from templates generated by Domino Global WorkBench, you will need to do the following:

1. Create a new project in the WorkBench.
2. Load the source template as the source.
3. Load any glossary.
4. Load the tagged template as the tagged db (using File - Add an existing Tagged Database).
5. Load each language database as target databases (using File - Add an existing Language Database).
6. Do not perform any update, rebuild or tagging options at this stage, but go straight to the synchronization pane.
7. Select the relevant synchronization options and synchronize.
8. The databases are now Synchronizer server task compliant.

Notes If design inheritance is used, the same synchronization that is performed on the database above should also be performed on the templates, so that they do not automatically overwrite the synchronization features added to Forms and so on. If you are refreshing/replacing the design manually, you can either again ensure the templates have been synchronized similarly, or repeat the process above after the refresh/replace (using the project you have created).

Using the Synchronizer configuration database

The Synchronizer configuration database is installed automatically when you install the Language Synchronizer server task. You use it for the following tasks:

- To view your synchronization sets.
- To enable or disable language synchronization within a synchronization set.
- To view logs of language synchronization activity.
- To view synchronization type and translation statistics for each synchronization set.
- To create new synchronization sets.
- To modify the language synchronization schedule for a synchronization set.
- To add databases to, or remove databases from, synchronization sets.

When you install the Language Synchronizer task on a Domino server, the Synchronizer configuration database is automatically installed into the Notes data directory of that server.

Viewing synchronization sets

Views in the database let you view:

- All synchronization sets, or only those that are enabled, and you can view them by owner or by set name.
- Synchronization logs by owner, by set name or by last-synchronized date/time.
- Translation and language synchronization statistics for each synchronization set.

Each synchronization set has:

- A synchronization set document which contains information about the set, including the synchronization schedule, which you can change if you want to.
- A synchronization log document which contains a history of language synchronization, including listings of any errors encountered. This document is created automatically when the set is synchronized for the first time.

Each database in a synchronization set has:

- A database information document, held as a response document to the synchronization set document, which contains information about the database.
- A database statistics document which reports the number of documents of each type in the database, how many have been authored in the database and how many require translation. This document is created automatically when the database is synchronized for the first time.

Enabling or disabling language synchronization

Documents will be synchronized within a synchronization set only if language synchronization has been enabled for that set. To enable language synchronization, select the relevant synchronization set document in the Synchronizer configuration database and click Enable. To disable language synchronization, select the relevant synchronization set document and click Disable.

Creating a synchronization set

There are three ways of creating a synchronization set:

1. By clicking the Configure Synchronization Set button in the WorkBench (see "Adding language synchronization features", Step 9, above).

2. By clicking Smart Setup in the Synchronizer configuration database. This opens a series of Setup dialog boxes where you enter information about the set you want to create. On completion, the necessary synchronization set and database information documents are created.
3. By clicking New Set in the Synchronizer configuration database. This creates a new synchronization set document. You must then add databases to the new set as explained below.

The main fields in a synchronization set document are:

Set name The name of the synchronization set. Typically this might be the title of the original source database from which the synchronized database(s) were built.

Current state Whether or not language synchronization is currently enabled for this synchronization set. Documents will be synchronized within a synchronization set only if language synchronization has been enabled. You can select Enabled or Disabled in the drop-down box to change the synchronization state.

Last edit details The person who last edited this document, and the date and time the last change was made. In order to edit a synchronization set document you must have editor access to the Synchronizer configuration database and to all of the databases in the synchronization set. The person who last edited the synchronization set document is regarded as the owner of the synchronization set.

Schedule information The frequency of language synchronization: a number of days, hours or minutes. If days are specified, a start time is also specified.

Last and next schedule times These fields show when the documents in the synchronization set were last synchronized and when they are next scheduled to be synchronized.

Scheduling the Language Synchronizer

To modify the language synchronization schedule for a synchronization set, edit the relevant synchronization set document:

1. In the Synchronizer configuration database, open the relevant synchronization set document.
2. Go to the Schedule tab.
3. Specify the frequency of language synchronization: a number of days, hours or minutes. If you specify days, you must also enter a start time. If you specify hours or minutes, synchronizing will start right away, assuming it is enabled.
4. Save the document.

Adding and removing databases

To add a database to a synchronization set, you must first deploy the database on a suitable Domino server and then create a database information document for it in the Synchronizer configuration database. To do this:

1. In the Synchronization configuration database, select the relevant synchronization set document.
2. Click Add Database.

3. Complete the fields in the new database information document.

The main fields in a database information document are:

Title A title for the database which will be used in the Synchronizer configuration database views. This does not have to be the same as the database's regular title.

Server The Domino server on which the database is located.

Filename The filename of the database. If it is not in the server's Notes data directory, enter the full pathname.

URL The URL of the database. This is needed if documents in the synchronization set use the Language SwitchBar. See "The Language SwitchBar", above. Note that the SwitchBar cannot be used in multilingual databases.

Language information Whether or not the database is multilingual, and the language(s) it supports. These fields are purely for your information. They are completed automatically if you use Smart Setup (see below). If you create a database information document from scratch, you can enter information or leave the fields empty.

4. Save the document.

To remove a database from a synchronization set, simply delete its database information document.

Moving databases

If you want to move a synchronized database to a different server you can do so, but you must then edit the relevant database information document to reflect the change. To do this:

1. In the Synchronizer configuration database, open the relevant database information document.
2. Edit the Server and Filename (or URL) fields.
3. Save the document.

Using synchronized databases

The following paragraphs contain advice about handling translatable documents in a synchronized database. If translation workflow is running in the databases, see also "Using translation workflow", below.

Creating translatable or global documents

You create documents in synchronized databases in just the same way as you create documents in any database. The only difference is that they contain (hidden) fields used by the Language Synchronizer. One of these hidden fields contains the state of the document (local, global or translatable). The initial (default) state depends on what was specified when the language synchronization features were added to the document's form. Often you will not want translatable or global documents to be viewed by users in other languages until they are complete. In that case, the initial state of the forms used to create the documents should have been set to 'Local'. If that was not done, you can use the 'ToLocal' agent to convert a translatable or global document to a local one. Provided you do this before the Language Synchronizer runs, the document will not be seen in other languages.

When you are finally ready to synchronize the document across languages, you can use either of the agents 'ToGlobal' or 'ToTranslatable' which are automatically added to every synchronized database, to change the type.

Note You can change the type of a document (global, local or translatable) only in the originally authored version of it.

Do not mix local global and translatable forms in the response hierarchy. For example, do not have a global document as a response to a local document because there will be no document for it to be a response to in other languages. This is especially true in multilingual databases, where only a single copy of global documents is created.

Note The language synchronization features added to a database by Domino Global WorkBench affect the forms in the database, not the views. Ensuring that the right documents are seen by the right users is a matter of creating the appropriate views when designing the database.

Translating translatable documents

This is a matter of editing (translating) the individual language documents that have been created by the Language Synchronizer.

Caution Unless you change them, the doc links in a translated document will continue to point to the documents specified by the original author. It is most likely that these documents will be in the original language. To replace a doc link first go to the new target document and choose Edit - Copy As Link - Document Link. Then delete the link in the document being translated and choose Edit - Paste to paste the new link into the document in its place. View links and URLs may also need to be changed.

Note Chapter 5 is concerned with the translation of the *design elements* of the database (part of the localization process), not its content.

Marking documents as translated

In order for a document to be recognized as translated, the field \$Lng_State must be set to 1. This should be done only when translation is complete (any extra workflow required to achieve this, for example approval states, is up to the database designer). It is advisable to add action buttons in an untranslated view or a translator specific form in order to facilitate this.

Translation workflow is provided to automate this task, as well as to provide approval states, tracking and automatic notification.

See "Forms for translators", above.

What the Language Synchronizer does

The Language Synchronizer does the following:

- It copies any new translatable documents to other languages and gives them the state 'Untranslated'. In the case of unilingual databases these copies will be put into the appropriate language databases.
- If the original version of an existing translatable document has changed the Language Synchronizer does one of the following:
 - If a companion translation document has the state 'Untranslated' or 'Translated', it changes the state to 'Modified'.
 - If a companion translation document has the state 'Modified', it does nothing.

- It copies new or changed global documents to other unilingual language databases. In a multilingual database, the Language Synchronizer simply adds the IDs for the other languages to the document's \$Language field.
- It does nothing to local documents, unless they were originally global or translatable. In the latter case it marks the corresponding documents as local.
- It removes all documents marked for deletion (see below).

The Language Synchronizer is a server task. See "Scheduling the Language Synchronizer", above, for information about running it.

Deleting documents

If you delete the original version of a translatable or global document, unlinked versions will remain in the database(s). If you delete a translatable or global document, not the original, it will be created again (from the original) the next time the databases are synchronized. For this reason an agent called 'MarkDeleted' is added to all synchronized databases. If you run this on the original version of a translatable or global document that you want to remove, it will set its state (\$Lng_State) to deleted. The next time the Language Synchronizer runs, it will then delete all the linked versions of the document, and finally the original itself.

Creating documents through agents

If you are creating new documents through script, macros or Java, and you want these documents to be synchronized, there are a number of fields you may need to be aware of (see "Fields in synchronized databases", above, for more information):

\$Language. Enter the language ID of the document being composed (generally the database language from the profile document).

\$Lng_UNID. The Language Synchronizer will set this up automatically, so you do not need to do anything.

\$Lng_SynchType. Enter the state of the form.

\$Lng_Original. Set to "1" as it will be is the original

\$Lng_State. Set to "1" as it is fully translated (being the original).

\$Lng_LastSynchronised. Set to Now.

\$Lng_LastModified. Set to Now.

Modifying synchronized documents through agents

Take care when modifying the original, authored version of a synchronized document, as a change to the last modification date will automatically tag the language versions as modified and therefore requiring attention.

Manually adding the subform

If a form has been built without a language synchronization subform (or perhaps the wrong one), the correct subform can be added to the form, thereby making it fully compatible with the Language Synchronizer (any previous language synchronization subform should be removed).

To do this:

1. Add the following fields to the form (they should be hidden), and give them appropriate default values:
 - \$Lng_SynchType
 - \$Lng_State
2. Add the subform LngSynchroniserMain.

3. If you want the initial synchronization type of the form to be user selectable (from both Notes and the Web), you should add a computed subform with the following formula:

`@If(!@IsNewDoc;"";@ClientType="Notes";"LngSynchroniserNotes";"")`

The easiest way to achieve this is to copy all the information from an existing form design with the settings you require.

Translation workflow

Translation workflow is a Domino application that builds on language synchronization technology to manage the process of document translation in Domino database(s). The databases typically comprise a Web site. You supply the application with information about the set of synchronized databases and the names of individuals responsible for performing steps in the translation process (for example, manager, translator, reviewer and approver). Workflow makes copies of documents ('working copies') for use during the translation process and informs the relevant people when each step in the process is completed. When a translation is finally approved, the contents of the workflow copy overwrite the original translation copy completing the translation process.

For example, suppose you have translation workflow operating in a set of synchronized databases that make up a Web site. New documents are created and edited as 'local' documents, which you have excluded from user views. When a document is final, its author makes it 'translatable'. Workflow then automatically guides it through the necessary translation processes. The basic workflow for such a document is:

- The author makes the original (source) document 'translatable' in a synchronized database.
- The Language Synchronizer, a server task, copies the new document to the other databases in the synchronization set. These copies are called 'translation documents'.
- For each translation document, the workflow agent creates a workflow tracking document in a 'translation workflow database'. The tracking document contains the names of the people involved in the translation process and the current status of the translation.
- The workflow agent (in the translation workflow database) makes a working copy of the new document, and sends a mail to the first person in the translation process.
- As each person completes his or her operation on the working copy, they update the status in the workflow tracking document, and the workflow agent sends a mail to the next person in the process.
- When the status finally is 'Approved', the workflow agent copies the text of the working copy into the original translation document and marks the latter as 'Translated'. It then deletes the working copy and the workflow tracking document.

A similar workflow applies when a document that has been translated is edited.

Just what documents can be viewed by users in synchronized databases is a matter for the database designer. Perhaps only 'Translated' documents will be displayed. Or perhaps all 'Translatable' documents will be displayed, regardless of whether or not they have been translated.

Before workflow can begin you must:

1. Prepare your source database, making sure that it contains the views you need.
2. Use the WorkBench to create one or more synchronized database(s) from your source database. This can be a single multilingual database or a set of unilingual databases.
3. Create a translation workflow database (from a template provided with Domino Global WorkBench).
4. In the translation workflow database create the following documents:
 - A workflow information document, which identifies the set of synchronized databases and establishes various defaults for the workflow.
 - Language information documents, one for each language you are translating into, which name the people involved in translations into that language (translators, reviewers and so on).

See Chapter 2: Recommended processes, "Adding translation workflow".

Workflow terminology

The following terms are used in the descriptions of translation workflow:

Language databases The synchronized databases that the translation workflow agents poll for translatable documents. These are databases built by Domino Global WorkBench.

Language information document A document in the translation workflow database that identifies the people who are responsible for translations into that particular language.

Source document A document that has been created and given the status 'Translatable'. A source document can be created in any language supported by the database(s) in workflow. Since creation it may have been edited.

Synchronization set A 'synchronization set' is a group of unilingual databases, or a single multilingual one, in which documents are being synchronized between languages. Translation workflow is set up in databases that comprise a synchronization set.

Translated document Either a translation document that has been through workflow or a source document in its original language. Translated documents are visible to users.

Translation document A copy of a source document. A translation document is created by the synchronizer for every supported language (other than the source language). To begin with these documents are untranslated. After going through translation workflow they become translated.

Translation workflow database The database that contains the documents and agents that control the translation workflow. People involved in the translation process use this database.

Workflow information document A document in the translation workflow database that identifies a set of synchronized databases and establishes various settings for the workflow. For example it specifies the number and names of the steps in the workflow process.

Workflow tracking document A document in the translation workflow database that controls the translation flow for a document being translated.

Working copy A copy of a source document that is being worked on by a translator. When it is marked Approved its contents will be copied to a translation document, and it will be deleted. The translation document will be marked as translated.

Workflow roles

When setting up translation workflow the webmaster can specify the number of review steps s/he wants to have in the workflow process. If two review steps are specified, the default steps are:

- Translation
- Graphics review
- Links review
- Approval

In this typical scenario, translation workflow expects there to be people in your organization with the following roles:

Webmaster The person responsible for the translation workflow database and for the synchronized databases that hold the documents being translated. The synchronized databases are typically a Web site. The webmaster sets up the initial workflow information document and ensures that approved translations are published to the Web. The webmaster is given the Notes role [Webmaster].

Translation manager The person responsible for translations into a language. The translation manager decides who will translate, review and approve translations for that language. Translation managers are given the Notes role [TranslationMngr].

Note In simple cases it is likely that the same person will carry out the webmaster role and (all) translation manager roles.

Translator The translator translates documents into his or her language, marking the document 'Ready for graphic review' when they have finished.

Graphic reviewer The graphic reviewer reviews the graphics in a translated document, editing them where necessary, and marking the document 'Ready for links review' when they have finished.

Links reviewer The links reviewer reviews the links (to other documents or views) in a translated document, changing them where necessary, and marking the document 'Ready for review' when they have finished.

Approver The approver reviews the translated document as a whole, editing it where necessary. The approver has the authority to mark a document as 'Approved' after all the required translation and review cycles are complete. If the webmaster specifies zero review steps, the graphics and links reviews will be skipped, and the workflow will go straight from translation to the approval stage.

Note There is no specific Notes role for translators, reviewers or approvers.

Preparing a database for translation workflow

Translation workflow is based on language synchronization, and you will find some general information about preparing databases for language synchronization in the section "Preparing a database for language synchronization", above. The information given here is specific to databases that will use translation workflow.

Views

Just what documents can be viewed by users in synchronized databases is a matter for the database designer: it is up to the designer to make sure that the database includes the views that are needed. Perhaps only 'Translated' documents will be displayed. Or perhaps all 'Translatable' documents will be displayed, regardless of whether or not they have been translated. Database users will generally need to see all documents in their language, and you may or may not want them to see translated documents whose source text (the text in the original language) has changed.

Local documents (`$Lng_SynchType="0"`) should usually be visible only to their owners. It is a good idea to make the initial state of new documents local -- you can do this when adding the language synchronization features. Local documents are ignored by the Language Synchronizer and the workflow agent. When an author is happy with the contents of a new document he or she can change it from local to global or translatable.

For people involved in the translation process you may want to include a view that shows working copies of documents (these people can alternatively access documents via mail links or through tracking documents in the translation workflow database). And you may want a view to show archived documents.

You can use the following fields to select and/or categorize documents:

- `$Lng_State`

"0" The document is untranslated, or, if translated, the translation is out of date. You may want to exclude these documents from user views altogether, or you may prefer to make them visible but flag them with a warning icon.

"1" The document is either translated or does not need translation. In the former case it is an approved translation of the source. In the latter case it is a copy of the source or the source itself.

"2" Workflow is in progress and this is a translator's working copy.

"3" The document is translated, but the original has changed. This is a transitory state. When workflow next runs, it changes to 0.

"4" A source document that has been marked for deletion. This is a transitory state. When the synchronizer next runs, this document will be deleted along with all its companion translation documents.

"5" The document is regarded as translated: it's a global or local document which doesn't need translation.

- `$Lng_SynchType`

"0" The document is local. It doesn't need translation.

"1" The document is global. It doesn't need translation.

"2" The document is translatable. It may or may not yet be translated.

- `$Language`

For example, it is "fr-FR" if the document is French as used in France or "en-US" if it is US English. See "Adding new languages" in Chapter 4 for more information about language identifiers.

Note `$Language` is used to select the document for language views. It says nothing about whether or not a document is translated. For example, to display French documents that have been translated `$Language` must be "fr-FR" *and* `$Lng_State` must be "1".

Setting up translation workflow

Translation workflow can operate only in synchronized databases. You create synchronized databases by building in the usual way, and then adding language synchronization features using the WorkBench. The Language Synchronizer must have been installed. See "Adding language synchronization features" and "Installing the Language Synchronizer", above.

To set up translation workflow:

1. Using Notes Release 5, create a translation workflow database on the Domino Release 5 server where the relevant Synchronizer configuration database is installed. Use the template DGWWFLOW.NTF. The translation workflow database contains documents that control the workflow process, agents to carry out workflow operations and tracking documents for the documents that are undergoing translation.
2. Give yourself, and each of the other people who will play a part in the translation process, editor access in each of the synchronized databases where the workflow will operate. Give yourself, and anyone else who will be managing the translation workflow database, 'Delete documents' rights.
3. Select the new translation workflow database in Notes and choose File - Database - Access Control. Give yourself, and anyone else who will be managing the translation workflow database, at least editor and preferably manager access. Make sure that 'Delete documents' access is checked.

Add the names of the other people who will play a part in the translation process, giving them editor access. You can have a different team of people for each language pair (for example, if your databases support English, Japanese, French, and German, you can specify different people for English to French, French to English, English to Japanese, and German to French tasks). If you want translation managers to be able to delete documents in the workflow database check the 'Delete documents' box.

4. Create a workflow information document in the new database. When you first open the database you will be prompted to do this. Otherwise choose Create - Workflow Information.

Note A translation workflow database cannot contain more than one workflow information document. If you want to run translation workflow in more than one set of synchronized databases, you must create a separate translation workflow database for each set.

The manager of the workflow database

The server that holds the Synchronizer configuration database

The name of the synchronization set in which workflow is to run

This section is filled automatically when you click ...

... this button

Workflow Information		
General	Workflow States	Options
Site Webmaster:	John Doe/NYC/XYZ Corp	
Synchronization server:	NYC_GENL/NYC/A/XYZ Corp	
Synchronization set name:	X12 sync set	
Language:	Database filename:	
French	fr1.nsf	
US English	en1.nsf	
Retrieve Synchronizer Data		

In the workflow information document supply the following information under the General tab:

- The name of the site webmaster. This is the manager of the translation workflow database, and usually also the manager of the synchronized databases. By default, this is the person who creates the workflow information document.
- The name of the synchronization set for which you are enabling translation workflow.

Click 'Retrieve Synchronizer Data' to enter automatically the languages supported by the synchronized database(s). This also checks that the databases exist and that they do contain synchronization features.

Workflow Information

General | Workflow States | Options

How many review cycles? → Levels of review:

Default names: Untranslated
Ready for translation
Ready for graphics review
Ready for links review
Ready for final approval
Approved

Here you can change the names of the workflow states to suit your process → Configured names:

Supply the following information under the Workflow States tab:

- (Optional) The number of levels of review that you require. The default is 2. The maximum is 5.
- (Optional) Names for the workflow states. The default names are: 'Untranslated', 'Ready for translation', 'Ready for graphic review', 'Ready for links review', 'Ready for review 1', . . . , 'Approved'. You should change these so that they are suitable for your particular translation and review process. The number of workflow states depends on the number of levels of review you specified.

Workflow Information

General | Workflow States | Options

Only send mails when next step is carried out by a different person? → Limit mail notifications?

One or more fields in the synchronized database documents that can be used for identification purposes in the workflow database → Hint fieldname:

Supply the following information under the Options tab:

- Whether or not to limit mail notifications. In general, on completion of each workflow operation, a mail is sent to the next person in the workflow process. However, this does not happen in cases where the next person is also the person responsible for the current step. If you set 'Limit mail notifications' to No a mail will be sent in all circumstances.
- The names of fields ('Hint fieldname') whose contents can be used to identify documents in views in the translation workflow database. You should enter one field name for each form that is used to create translatable documents. If the hint field name is the same for several forms you need enter it only once: the workflow application always uses the first matching fieldname it finds.

For example, suppose you have fields called 'Subject' and 'Title' on one form, a field called 'Subject' on a second form, and a field called 'Title' on a third form. Then, if you enter Title and Subject as hint fieldnames, you will see, as identifying text in the workflow database's Comment column, the Title of documents created from the first form, the Subject of documents created from the second form and the Title of documents created from the third form.

Click Save and Close. An agent now automatically creates language information documents, one for each language. These documents specify who will be the translation managers for translations into the language (one for each other language supported by the workflow). They also specify the default translators and reviewers for translations into the language.

The person named as site webmaster is automatically assigned the role [Webmaster].

5. Edit each of the language information documents which were created automatically at the end of Step 4. You will see them in the Setup view under Language Information.

The screenshot shows a form titled "Language Information" with the following fields and callouts:

- Language Information:** A dropdown menu with "French" selected. Callout: "This document relates to translations into this language. The field is filled in automatically when the document is created".
- Language Database:** A text field containing "workflow\lfr1.nsf".
- Target Language:** A dropdown menu with "French" selected.
- Source language(s):** A dropdown menu with "US English" selected. Callout: "Other language(s) in the synchronized database(s)".
- Language ID(s):** A text field containing "en-US".
- Enable Translation Manager review?:** A radio button set to "No". Callout: "If set to Yes the translation manager(s) will be notified by mail about new translatable documents so that they can review, and change if necessary, the work assignments".
- Workflow Assignments:** A button. Callout: "Click to assign people to translation tasks for the target language specified above".

Each language information document relates to translations into one (target) language. In each language information document:

- Select Yes or No for translation manager review. If this is set to Yes the translation manager will receive a mail whenever a new, translatable document is created for this language pair. He or she can then review the default job assignments, changing them if necessary, finally marking the document 'Ready for translation'.

- Click Workflow Assignments to display the Select Work Assignments dialog.

- Select a source language.
- Click Get Assignments to fill the dialog fields with the names of the people (if any) who currently are the translation manager, translator, reviewer, approver and so on for this source/target language pair.
- Enter (or change) the names of people to fill the various workflow roles for the translation of documents from the selected source language into the target language.

The people you assign here must be in the database's access control list (ACL). You can leave any of these fields blank if you want to. In that case, workflow will assign the task in question to the translation manager for the relevant language pair or, if no translation manager is specified, to the site webmaster.

Translation managers can initially be assigned only by the person named as site webmaster in the workflow information document. They are given the Notes role of [TranslationMngr] automatically when Set Assignments is clicked.

In general, when a workflow step is marked as complete, a mail is automatically sent to the person responsible for the next step. However, if that person is also responsible for the current workflow step, and 'Limit mail notifications' is set to Yes in the workflow information document, no mail is sent.

Once a language information document has been created, anyone with the [Webmaster] or [TranslationMngr] role can make changes to the work assignments.

Note These are *default* work assignments: if 'Translation manager review' has been set to Yes (above), the translation manager will be given the opportunity to change the translator, reviewer and approver assignments for each individual document.

- Click Set Assignments to copy the contents of the dialog fields back to the language information document.
- If necessary Select another source language and repeat the selection steps.
- Click OK to finish.

People assigned to the various tasks can progress the workflow to the next stage when they have finished their particular operation. Users assigned to the [TranslationMngr] role can progress documents from any stage of the workflow to the next. Users assigned to the [Webmaster] role can perform all tasks.

6. Set a schedule for the translation workflow agent:
 - In the translation workflow database, choose View - Agents.
 - Open 'Workflow'.
 - Under 'When should this agent run?' select one of the 'On schedule' options. Usually this will be 'On Schedule Daily' or 'On Schedule More Than Once a Day'.
 - Click Schedule and enter the schedule information. At a minimum you should run workflow agents each time the Language Synchronizer has completed its operation. For information about the start and stop times of the Language Synchronizer, tab to 'History' in the relevant synchronization set document (in the Synchronizer configuration database).

Notes The workflow agent can also be run manually from the Actions menu. In order for it to operate correctly, the person who runs it (or the last person who modified its schedule) must have the role [Webmaster] or [TranslationMngr] and document creation and deletion rights in all databases in the synchronization set.

Messages from the translation workflow agent are stored in log documents. You can view these in the Log Documents view.

7. Enable the [Webmaster] role for the server where the translation workflow database is located:
 - Select or open your translation workflow database.
 - Choose File - Database - Access Control.
 - On the Basics panel select the ACL entry for the server where the database is located.
 - In the Roles listbox, select [Webmaster] to enable the role for this server.

Using translation workflow

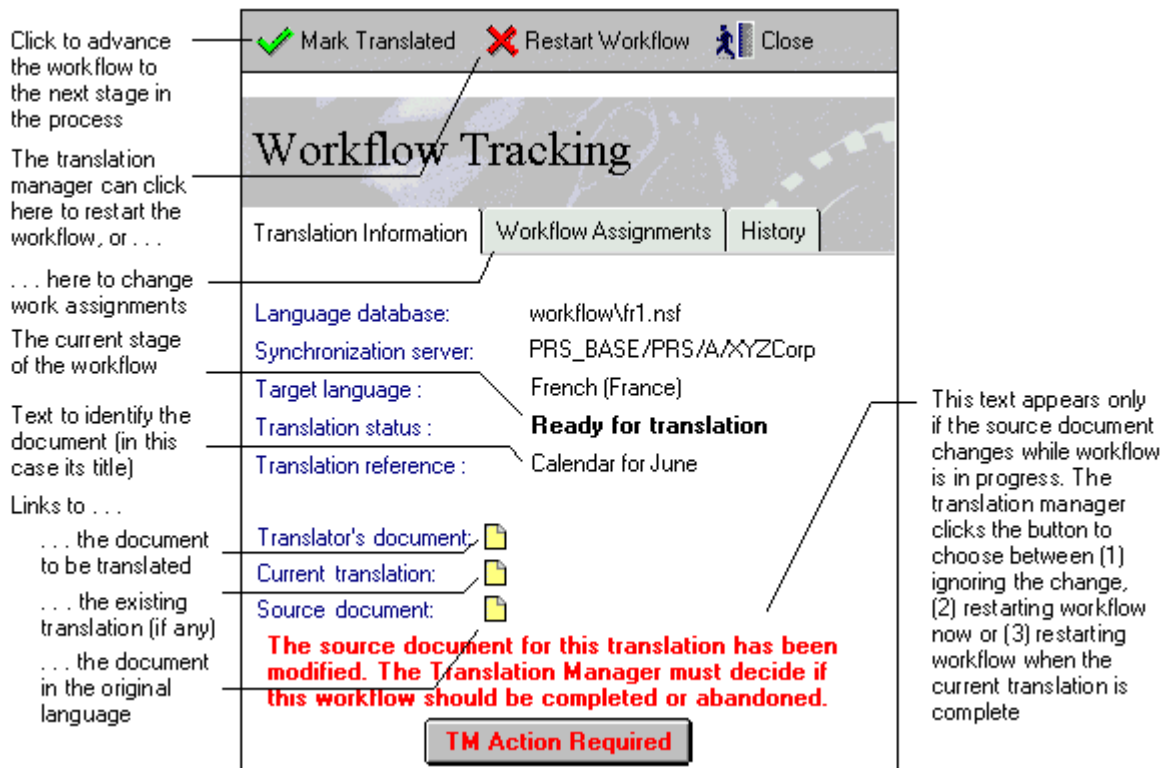
The following paragraphs explain how to use translation workflow once it has been set up (for more detail see "How translation workflow works", below). For this illustration it is assumed that:

- There are three unilingual synchronized databases in which workflow is running: US English, Japanese and German.
- The default state for new documents is 'local' (this is set when adding the language synchronization features to the databases).
- The database views are set up so that local documents are visible only to their authors and translatable but untranslated documents are not seen by any users. See "Views", above.
- The workflow stages are the default ones: 'Untranslated', 'Ready for translation', 'Ready for graphic review', 'Ready for links review', 'Ready for final approval' and 'Approved'.
- Translation manager review is set to Yes, allowing the translation manager to review the default job assignments.

The process is:

1. An author creates a new document, the source document, in (say) the German database. The document has the synchronization state of 'local'. The author edits the new document as necessary. When it is complete s/he selects it in a suitable view and chooses Actions - ToTranslatable. This makes the document 'translatable'. As a newly created document it is also automatically marked 'translated'. It will now be seen by users of the German database.
2. When the Language Synchronizer next runs it detects the new translatable document in the German database and copies it to each of the other databases. These copies are 'translatable' but they are 'untranslated'. They are not seen by users.
3. When the workflow agent runs it detects the new translatable documents in the US English and Japanese databases. It does the following:
 - Creates a working copy of each new translatable document in the US English and Japanese databases. Working copies are not normally seen by users. However, the database designer may have created a view to display them for translation managers.
 - Creates a tracking document in the translation workflow database.
 - Mails the translation managers for German to US English and for German to Japanese informing them of the new documents that need translation. Each mail contains a doc link to the relevant tracking document which in turn contains links to the working copy and to the source document. (There is also a link to the current translation document, but in this case, as we are dealing with a new document, the text of the translation document will be the same as that of the source.)
4. The translation managers review the work assignments in the tracking documents, and change them if they want to. Finally, they give the tracking documents the status 'Ready for translation' either:
 - By selecting the workflow tracking document in the My Work - By Status view and clicking 'Phase Completed'. By using this method, several workflow tracking documents can be selected and processed at once.
 - or
 - By opening the workflow tracking document and clicking 'Mark Ready for Translation'. This method of advancing the status of a tracking document provides an opportunity to add a comment for the next person in the workflow.

As soon as the status has been changed, the workflow application mails the translators for German to US English and for German to Japanese informing them of the new documents that need translation. Each mail contains a doc link to the relevant tracking document which in turn contains links to the working copy and to the source document.



Note If you do not want to translate a source document into a particular language, simply progress the workflow through all its stages (for example by using the Phase Completed button in the My Work - By Status view) without doing any translation. The translation workflow agent will then mark the document as translated, and workflow will stop. Even though you do not translate, however, you may need to make changes to document links in the translation document (see Step 7, below).

- The translators translate the working copies. Finally, they give the corresponding tracking documents the status 'Ready for graphic review' using one of the methods outlined in Step 4, above.

As soon as the status has been changed, the workflow application mails the graphics reviewers for German to US English and for German to Japanese informing them of the newly translated documents that need graphic review. Each mail contains a doc link to the relevant tracking document which in turn contains links to the working copy and to the source document.

- The graphics reviewers review the working copies. Finally, they give the corresponding tracking documents the status 'Ready for links review' using one of the methods outlined in Step 4, above.

As soon as the status has been changed, the workflow application mails the links reviewers for German to US English and for German to Japanese informing them of the documents whose links need reviewing. Each mail contains a doc link to the relevant tracking document which in turn contains links to the working copy and to the source document.

7. The links reviewers review the working copies. Doc links in a translated document will continue to point to untranslated documents in the source database unless they are replaced during one of the workflow steps. The links review step is designed for this function. To replace a doc link first go to the new target document and choose Edit - Copy As Link - Document Link. Then delete the link in the document being translated and choose Edit - Paste to paste the new link into the document in its place. View links and URLs may also need to be changed.

Finally, the links reviewers give the corresponding tracking documents the status 'Ready for final approval' using one of the methods outlined in Step 4, above.

As soon as the status has been changed, the workflow application mails the approvers for German to US English and for German to Japanese informing them of the documents which need final approval. Each mail contains a doc link to the relevant tracking document which in turn contains links to the working copy and to the source document.

8. The approvers review the working copies. They then either:
 - Give the corresponding tracking documents the status 'Approved' using one of the methods outlined in Step 4, above.
 - or
 - Click a Reject button on the tracking document. This resets the workflow to one of the previous stages (which the approver specifies). After clicking the Reject button the approver can add a comment to explain why s/he has not approved the translation.
9. When the workflow agent next runs:
 - It copies the contents of the now translated working copies into the untranslated documents in the US English and Japanese databases.
 - It marks the now translated documents 'translated' so that they appear in user views.
 - It marks the workflow tracking documents as inactive. The Completed/ Abandoned Items view in the translation workflow database shows inactive workflow tracking documents. You can delete them if you wish.
 - It deletes the working copies.

The workflow is now complete.

Note Because translation workflow is an extension of language synchronization, most of the advice given for synchronized databases also applies to databases operating translation workflow. See "Using synchronized databases", above.

Canceling translation workflow and deleting workflow documents

Do not abandon workflow or delete workflow documents except by the approved methods noted below.

- To delete a source document, select it and mark it for deletion by running the agent MarkDeleted. When the Language Synchronizer next runs, it deletes the document *and any translation documents created from it*. If workflow is in progress, the workflow agent will cancel it when it next runs.

Caution Do not delete a source document until you are sure that its companion translation documents are not needed or have been saved.

- Do not delete translation documents in isolation from their originating source document. Instead, run MarkDeleted, as indicated above, to delete the source document: the translation documents will automatically be removed by the Language Synchronizer.

If you do delete a translation document without deleting its source document, the Language Synchronizer will re-create it from the source document when it next runs. If workflow is active, and the workflow agent runs *before* the Synchronizer has re-created the translation document, the workflow agent will report an error, mail the working copy to the translation manager, and mark the workflow tracking document as inactive. Once the Language Synchronizer has re-created the translation document, the workflow agent will detect it, see that it has no tracking document, and restart workflow.

- Do not delete working copy documents. If you want to abandon the current translation workflow, use one of the following methods:
 - To restart workflow, open the relevant workflow tracking document and click 'Restart Workflow' (the button displays only if you have the role [Webmaster] or [TranslationMngr]). When it next runs, the workflow agent will delete the working copy and translation workflow will start again from the beginning.
 - To cancel workflow in all languages for a particular source document, first create a new source document with type global or local (assuming you want to keep the original source text) and copy the contents of the original source document into the new document. Then delete the original source document using the MarkDeleted agent.
 - If you do not want to translate a source document into a particular language, simply progress the workflow through all its stages (for example by using the Phase Completed button in the My Work - By Status view) without doing any translation. The translation workflow agent will then mark the document as translated, and workflow will stop.

Note Even though you do not translate, you may need to make changes to document links in the translation document.

Caution If you want to save any translation work done so far, do so before you cancel or restart the workflow. To do this, go to the working copy by clicking the 'Translator's document' link in the workflow tracking documents. Either copy and paste the text to another document or choose Actions - Forward to mail it to a suitable person.

- Do not delete active workflow tracking documents. If you want to abandon the current translation workflow, use one of the methods described above. When it next runs, the workflow agent will mark the existing tracking document 'Inactive'. Provided you have the role [Webmaster] or [TranslationMngr] you can then delete it from the Completed/Abandoned Items view.

How translation workflow works

There are several possible scenarios for the sequence of events once workflow has been established:

- A new source document is created.
- A source document is updated (earlier workflow is complete).
- A source document is updated (earlier workflow is not complete).
- A source document is deleted.

Workflow when a new document is created

The following paragraphs describe the translation workflow sequence when a new document is created, and show how the values of the key fields change.

1. The original document is created as a Local document in a working database.

\$Lng_SynchType=0, Local

\$Lng_State=5, Not a translatable document

The Language Synchronizer server task does not touch Local documents. Only Local and Global documents have \$Lng_State=5. By inference, any document with \$Lng_State=5 can be considered 'translated' because it has been authored in its native language.

2. When the original document is complete its state is changed by its author to Translatable. This becomes the source document for translation workflow:

\$Lng_SynchType=2, Translatable

\$Lng_State=1, Translated

Source documents do not need translation and are therefore not themselves part of the translation workflow.

3. The Language Synchronizer runs. It makes a copy of the source document for each of the other languages. These are translation documents and they require translation:

\$Lng_State=0, Translatable but untranslated

This value can be used in view selections to exclude the original versions of documents that are in workflow from being seen by database users. Alternatively, you may want users to be able to access these documents, even though they are untranslated. In that case you can use the value to display a warning icon to indicate that the documents are being translated.

4. When the Language Synchronizer has finished running, the workflow agent runs. It does the following:

- Detects new translation documents (\$Lng_State=0) that are not already in workflow (have no workflow tracking document).
- For each of these it makes a working copy of the corresponding source document: the working copies are the documents that the translators will translate and reviewers will review. The working copies are held in the relevant language database and they are marked as 'Workflow in progress':

\$Lng_State=2, Workflow in progress

This value can be used in view selections to display working documents for translators and reviewers, and to prevent working documents from being seen by other database users.

- Creates workflow tracking documents in the translation workflow database, one for each translation document.

Note You can use \$Lng_State to select documents for views and/or replication. A value of 0 means that a document is either untranslated or out of date (its source document has changed). A value of 2 means that it is a working document. A value of 1 or 5 means it is current and translated. A value of 3 or 4 means the document is translated but its source document has been modified or is scheduled for deletion. A value of 6 means the document is no longer current and has been archived.

Each workflow tracking document contains:

- Doc links to the working copy, to the source document, and to the current translation document. In this case, as we are dealing with a new document, the text of the translation document will be the same as that of the source (when workflow is complete, the contents of the working copy will be copied into the translation document, making it translated).
 - The language into which the working copy needs to be translated.
 - The names of the people who will carry out the various tasks: translation, review and so on. Default names are taken from the relevant language preferences document.
 - The current status of the document (ready for translation, translated, ready for review, approved and so on).
5. If 'Translation manager review' has been set to Yes in the language information document, a mail is sent to each of the relevant translation managers with a doc link to the appropriate workflow tracking document. Once a translation manager has made any necessary changes to the assignments in a workflow tracking document, s/he clicks Mark Reviewed to mark it as 'Ready for translation' (this sets the value of a workflow state field in the workflow tracking document).

If 'Translation manager review' has been set to No, each workflow tracking document is automatically marked as 'Ready for translation'.

Notes An agent allows the translation manager to mark several documents at once. When setting up translation workflow, you can tailor the names of the workflow states (for example 'Ready for translation') to suit your particular process.

6. A mail is sent to the translators (one for each language). It contains a doc link to the relevant tracking document which in turn contains links to the working copy and to the source document.
7. Each translator translates the working copy. When translation is complete, the translator either:
- Selects the workflow tracking document in the My Work - By Status view and clicks 'Phase Completed' to give it the next workflow state, for example 'Ready for graphic review', or
 - Opens the workflow tracking document and clicks the 'Mark . . . ' button (the button label varies according to the workflow state) to give the document the next workflow state. This method of advancing the state of a tracking document provides an opportunity to add a comment to the history field of the tracking document.

By using the first of these methods, several workflow tracking documents can be selected and processed at once, but comments cannot be added.

When the workflow agent next runs it sends a mail with appropriate doc links to the next person in the translation process.

8. Step 7 is repeated for the other tasks in the workflow. Finally, the approver marks the tracking document with the workflow state 'Approved'.
9. When the workflow agent next runs it:
- Identifies tracking documents in the translation workflow database with state 'Approved'.

- Checks that the \$Lng_State of the translation documents (the ones created in Step 3) remains 0. If the source document (the one created in Steps 1 and 2) has been changed this value will be 3. If the source document has been deleted its translation documents will no longer exist. See "Workflow when a document is updated (earlier workflow not complete)" and "Workflow when a document is deleted", below, for explanations of workflow in those cases.
- Overwrites the translation documents (the ones created in Step 3) with the translated and approved working copies.

Note In fact, the contents of the translated working copy are copied back into the original untranslated document. This preserves the Notes UNID (unique ID) of the document, which is important for the language synchronization process.

Caution If you are implementing your own workflow in this way, it is important that you preserve the original value of the \$Lng_LastSynchronised field during this operation.

- Marks these documents as translated:
\$Lng_State=1, Translated
- Deletes the working copies from the language databases.
- Marks the workflow tracking documents as inactive. The Completed/Abandoned Items view in the translation workflow database shows inactive workflow tracking documents. You can delete them if you wish.

Workflow when a document is updated (earlier workflow complete)

The following paragraphs describe the workflow sequence when a document is updated. The document has translations which were created by the new document workflow process (described in the previous section). That workflow process is now complete.

1. The source document is edited. Its translation state remains Translated:
\$Lng_State=1, Translated
2. The Language Synchronizer runs. It detects a change in the source document and marks the corresponding translation documents (not the source document) as Modified:
\$Lng_State=3, Modified
3. When the Language Synchronizer has finished running, the workflow agent runs. It does the following:
 - Detects translation documents whose source document has changed (these are translation documents with \$Lng_State=3) and which are not already in workflow (have no workflow tracking document). Changes the state of these documents to 'Translatable but untranslated':
\$Lng_State=0, Translatable but untranslated

This value can be used in view selections to exclude the original versions of documents that are in workflow from being seen by database users. Alternatively, you may want users to be able to access these documents, even though they are untranslated or out of date. In that case you can use the value to display a warning icon to indicate that the documents are being translated (or retranslated).

- For each of these it makes a working copy of the corresponding source document: the working copies are the documents that the translators will translate and reviewers will review. The working copies are held in the relevant language database and they are also marked as 'Workflow in progress':

\$Lng_State=2, Workflow in progress

This value can be used in view selections to display working documents for translators and reviewers, and to prevent working documents from being seen by other database users.

- Creates workflow tracking documents in the translation workflow database, one for each translation document. Each workflow tracking document contains doc links to the relevant working copy of the edited source document and to the relevant existing translation.
4. If 'Translation manager review' has been set to Yes in the language preferences document, a mail is sent to the translation manager with doc links to the workflow tracking documents. Once the translation manager has made any necessary changes to the assignments in each workflow tracking document, s/he marks it as 'Ready for translation'.

If 'Translation manager review' has been set to No, each workflow tracking document is automatically marked as 'Ready for translation'.

'Ready for translation' is the value of a workflow state field in the workflow tracking document.

5. A mail is sent to each translator. It contains a doc links to the working copy, to the source document, and to the current translation document.
6. Each translator translates the working copy. When translation is complete, the translator either:
 - Selects the workflow tracking document in the My Work - By Status view and clicks 'Phase Completed' to give it the next workflow state, for example 'Ready for graphic review', or
 - Opens the workflow tracking document and clicks the 'Mark . . . ' button (the button label varies according to the workflow state) to give the document the next workflow state. This method of advancing the state of a tracking document provides an opportunity to add a comment to the history field of the tracking document.

By using the first of these methods, several workflow tracking documents can be selected and processed at once, but comments cannot be added.

When the workflow agent next runs it sends mails with appropriate doc links to the reviewer.

7. Step 6 is repeated for the other tasks in the workflow. Finally, the approver marks the tracking document with the workflow state 'Approved'.
8. When the workflow agent next runs it:
 - Identifies tracking documents in the translation workflow database with state 'Approved'.
 - Checks that the \$Lng_State of the translation documents remains 0, as set in Step 3. If the source document (the one edited in Step 1) has been changed again this value will be 3. If the source document has been deleted its translation documents will no longer exist. See the following sections for explanations of workflow in those cases.

- Overwrites the translation documents with the newly translated and approved working copies.

Note In fact, the contents of the translated working copy are copied back into the original untranslated document. This preserves the Notes UNID (unique ID) of the document, which is important for the language synchronization process.

Caution If you are implementing your own workflow in this way, it is important that you preserve the original value of the \$Lng_LastSynchronised field during this operation.

- Marks these documents as translated:
\$Lng_State=1, Translated
- Deletes the working copies from the language databases.
- Marks the workflow tracking documents as inactive. The Completed/Abandoned Items view in the translation workflow database shows inactive workflow tracking documents. You can delete them if you wish.

Workflow when a document is updated (earlier workflow not complete)

The following paragraphs describe the workflow sequence when a document is updated and an earlier workflow process (new document or update) is still in progress.

1. The source document is edited. Its translation state remains Translated:
\$Lng_State=1, Translated
2. The Language Synchronizer runs. It detects a change in the source document and marks the corresponding translation documents (not the source document) as Modified:
\$Lng_State=3, Modified
3. When the Language Synchronizer has finished running, the workflow agent runs. It does the following:
 - Detects translation documents whose source document has changed (these are translation documents with \$Lng_State=3) *and* which are in workflow (have a current workflow tracking document). Changes the state of these documents to 'Translatable but untranslated':
\$Lng_State=0, Translatable but untranslated
 - Issues a warning message indicating that the source document for a document currently being translated has changed. The message is written to the server log, ALOG.NSF.
 - For each of these documents it sends a mail to the relevant translation manager and translator alerting them to a change in the source. The mail includes the text of the translator's current, part-completed working copy.

Caution Do not delete these mails until you are sure that the part-translated text they contain is no longer needed.

 - Adds the following text to the bottom of the relevant workflow tracking document(s):

The source document for this translation has been modified. The Translation Manager must decide if this workflow should be completed or abandoned.

TM Action Required

4. Clicking the button in the workflow tracking document gives the translation manager the following choices:
 - To continue the current workflow and ignore the update. In this case, the existing documents are left unchanged, as is the workflow state field in the workflow tracking document, and the process continues at Step 5.
 - To restart the workflow immediately using the updated text. In this case, the half-completed working copy is deleted and the workflow tracking document is marked as inactive. When the workflow agent next runs, it finds a translatable language document (\$Lng_State=0) that does not have an active workflow tracking document, and so restarts workflow from the beginning.
 - To continue the current workflow through to completion and then restart workflow for the updated source document. At the end of the current workflow, the translation will be marked as untranslated (\$Lng_State=0) so that workflow restarts. The first translation will be available to the person who translates the update.
5. If the translation manager has decided that the current workflow should continue, each translator translates the working copy. When translation is complete, the translator either:
 - Selects the workflow tracking document in the My Work - By Status view and clicks 'Phase Completed' to give it the next workflow state, for example 'Ready for graphic review', or
 - Opens the workflow tracking document and clicks the 'Mark . . . ' button (the button label varies according to the workflow state) to give the document the next workflow state. This method of advancing the state of a tracking document provides an opportunity to add a comment to the history field of the tracking document.

By using the first of these methods, several workflow tracking documents can be selected and processed at once, but comments cannot be added.

When the workflow agent next runs it sends mails with appropriate doc links to the reviewer.

6. Step 5 is repeated for the other tasks in the workflow. Finally, the approver marks the document with the workflow state 'Approved'.
7. When the workflow agent next runs it:
 - Identifies tracking documents in the translation workflow database with state 'Approved'.
 - Checks that the \$Lng_State of the translation documents remains 0, as set in Step 3. If the source document (the one edited in Step 1) has been changed again this value will be 3 and the process described here will begin again. If the source document has been deleted its translation documents will no longer exist. See the following section for an explanation of workflow in the case of a deletion.
 - Overwrites the original translation documents with the newly translated and approved working copies.

Note In fact, the contents of the translated working copy are copied back into the original untranslated document. This preserves the Notes UNID (unique ID) of the document, which is important for the language synchronization process.

Caution If you are implementing your own workflow in this way, it is important that you preserve the original value of the \$Lng_LastSynchronised field during this operation.

- Marks these documents as translated:
\$Lng_State=1, Translated
- Deletes the working copies from the language databases.
- Marks the workflow tracking documents as inactive. The Completed/Abandoned Items view in the translation workflow database shows inactive workflow tracking documents. You can delete them if you wish.

Workflow when a document is deleted

If a translatable document is deleted before workflow starts or after workflow has completed, then it will simply never be processed by the workflow agent. The source and its translations will be removed by the synchronizer before the main part of the workflow agent runs.

The process for deleting or archiving a translatable document while workflow is in progress is as follows:

1. The original, source document is selected and marked for deletion by running the agent MarkDeleted. Its translation state is changed to:
\$Lng_State=4, Deleted
 - Note** This is the only permitted way to delete a translatable or global document in a database operating translation workflow. If these documents were to be deleted in the usual way, their corresponding translation documents would remain undeleted.
2. The Language Synchronizer runs. It detects that the original document has been marked for deletion and deletes the corresponding translation documents and then deletes the original source document.
3. The workflow agent runs. It does the following:
 - Detects any workflow tracking documents whose corresponding translation documents no longer exist.
 - For each of these documents, sends a mail to the relevant translation manager and translator informing them that the original document has been deleted. The mail contains the text of the current, part-translated working copy.
 - Caution** Do not delete these mails until you are sure that the part-translated text they contain is no longer needed.
4. The workflow agent deletes the working copy and marks the workflow tracking documents as inactive. The Completed/Abandoned Items view in the translation workflow database shows inactive workflow tracking documents. You can delete them if you wish.

Replicating synchronized and translation workflow databases

In order to limit the opportunities for errors and replication conflicts, you should avoid working with replicas of synchronized databases, or databases in which translation workflow is operating. In particular:

- When making changes in the Synchronizer configuration database or the translation workflow database, use the original database, not a replica.
- Do not work in a replica of a multilingual synchronized database.

- You can work in a replica of a unilingual synchronized databases, provided you observe the following cycle of events:
 - Changes are made to documents in the replica. The changes will typically result from translation work.
 - Changes to documents in the replica are temporarily forbidden.
 - Changes are replicated back to the original database.
 - The Language Synchronizer runs.
 - Changes made by the Language Synchronizer are replicated back out to the replica.
 - Changes to documents in the replica are once more allowed.
- You can work in a replica of a unilingual synchronized databases in which translation workflow is running, provided you observe the following cycle of events:
 - Changes are made to documents in the replica. The changes will typically result from translation work.
 - Changes to documents in the replica are temporarily forbidden.
 - Changes are replicated back to the original database.
 - The Language Synchronizer runs.
 - The translation workflow agent runs.
 - Changes made by the Language Synchronizer and the translation workflow agent are replicated back out to the replica.
 - Changes to documents in the replica are once more allowed.

Note With version 5.0.5 of Domino Global WorkBench, unilingual synchronized databases can be placed on *any* server; they are no longer restricted to a single directory on one server. This should limit the need for replication, as the databases can be put on servers in the most geographically suitable locations.

Appendix

Error messages

The Appendix lists the Domino Global WorkBench error messages you may encounter.

Notes In addition to the messages listed below, you may also see messages that originate from the Notes API. For information about these, please refer to the Notes documentation.

If you have problems restarting the WorkBench after a crash, you should restart your machine. That will ensure that the Notes C API run time is shut down properly.

WorkBench error and warning messages

Below is an alphabetical listing of the WorkBench warning and error messages. An explanation is given for each message and, where appropriate, suggested action.

Error and warning messages appear in the tabbed log window at the bottom of the WorkBench display. Most will be accompanied by the name of the particular design element that caused the problem, but some are more general summary messages. Summary messages are usually preceded by more specific ones.

A field with a bad data type has been encountered

Meaning: The internal structure of the specified field has been corrupted.

Solution: Try opening and resaving the design element in the source database, and then update or rebuild the tagged and language databases. If that does not work you may have to re-create the design element from scratch.

An error occurred while accessing the registry. This new project database will not be stored.

Meaning: Problems encountered reading or writing to the Windows registry.

Solution: If you are using Windows NT you may not have the security access you need. You should have Administrator access when running the WorkBench.

Anomaly encountered. A term marked as tagged has no tag ID:

Meaning: A term was encountered with tag delimiters but no tag ID.

Solution: Use the WorkBench to rebuild the tagged database.

Bad color format for the icon:

Meaning: Errors encountered while processing the database icon.

Solution: Try opening and resaving the database icon in the source database (choose View - Design - Other), and then update or rebuild the tagged and language databases. If that does not work you may have to re-create the icon from scratch.

Bad field type found

Meaning: The internal structure of the specified field has been corrupted.

Solution: Try opening and resaving the design element in the source database, and then update or rebuild the tagged and language databases. If that does not work you may have to re-create the design element from scratch.

Bad header format

Meaning: The internal structure of this design element has been corrupted.

Solution: Try opening and resaving the design element in the source database, and then update or rebuild the tagged and language databases. If that does not work you may have to re-create the design element from scratch.

Bad icon data:

Meaning: Errors encountered while processing the database icon.

Solution: Try opening and resaving the database icon in the source database (choose View - Design - Other), and then update or rebuild the tagged and language databases. If that does not work you may have to re-create the icon from scratch.

Bad view format type

Meaning: The internal structure of a view has been corrupted.

Solution: Try opening and resaving the view in the source database, and then update or rebuild the tagged and language databases. If that does not work you may have to re-create the view from scratch.

Can't add term to glossary.

Meaning: The WorkBench was unable to create the term in the glossary database.

Solution: Check that the glossary is not read only, that you have the necessary rights to it and that there is space available on your hard disk. Open the glossary in Notes to check that the file has not been corrupted. If the glossary is held on a CD it will usually be read only: you must copy it to a hard disk before the WorkBench can use it. You may want to create a new glossary to receive new terms (specify it on the Basics panel of the Tagging Options dialog box).

Cannot build the language database because all views and folders are currently excluded

Meaning: The WorkBench was unable to build a language database.

Solution: A Notes database must contain at least one view or folder. Check that at least one view or folder has been selected for building (in the Name pane) and try again.

Cannot delete the project <project name>. The folder may be in use.

Meaning: You can delete a project by right-clicking the project icon (on the left hand frame of the WorkBench) and choosing Remove Project. However, you can only do so if the project is not loaded.

Solution: Unload the project and try again.

Cannot delete the folder <folder name>. The folder may be in use.

Meaning: You can delete a folder, and the projects it contains, by right-clicking the folder icon (on the left hand frame of the WorkBench) and choosing Remove Folder. However, you can only do so if none of its projects is currently loaded.

Solution: Unload the project(s) and try again.

Can't load HTML translation library. Copying text.

Meaning: The HTML translation library could not be found. HTML will therefore be copied, untagged, to the tagged database. If the problem persists you will have to translate the HTML text separately, in each of the language databases.

Solution: Your computer may be low on memory: close some applications and then try rebuilding the tagged database. Check that the file LTHTF.DLL is present in the Domino Global WorkBench program directory. If it is not, you should reinstall Domino Global WorkBench.

Can't load LotusScript translation library. Copying script.

Meaning: The LotusScript filter DLL could not be found. LotusScript will therefore be copied, untagged, to the tagged database. If the problem persists you will have to translate the LotusScript text separately, in each of the language databases.

Solution: Your computer may be low on memory: close some applications and then try rebuilding the tagged database. Check that the file LTLFSF.DLL is present in the Domino Global WorkBench program directory. If it is not, you should reinstall Domino Global WorkBench.

Cannot rename the item <name>. The source or destination may be in use.

Meaning: You can rename a project or folder by right-clicking its icon (on the left hand frame of the WorkBench) and choosing Rename. However, you can only do so if the new name is not already in use and the project (in the case of a folder any project it contains) is not loaded.

Solution: Choose a new name or unload project(s) and try again.

Can't run HTML filter

Meaning: The HTML filter cannot be invoked from Windows.

Solution: Your computer may be low on memory: close some applications and then try rebuilding the tagged database. Check that the file LTHTF.DLL is present in the Domino Global WorkBench program directory. If it is not, you should reinstall Domino Global WorkBench.

Can't translate LotusScript. Copying script.

Meaning: This message comes from the LotusScript filter used for tagging. The script cannot be tagged, it will be copied unchanged to the tagged database.

Solution: This is a summary message. It will be preceded by more specific messages that pinpoint the problems.

Compile error (script, formula)

Meaning: The LotusScript or formula could not be compiled. You may see this message when building language databases.

Solution: The most likely cause of this error is that some translated text in the script or formula has made its syntax invalid. Check in the language database that the translation does not include any special characters that are used in formulas or script, for example double quotes (") or semicolons (;). Check that no program keywords have been translated. Correct the translation that is causing the problem by editing the appropriate language document in the glossary.

If this does not work, try recompiling the LotusScript using Domino Designer. Locate the design element where the error occurred, and open the script that has the compile error. Make a trivial change, for example add a space and then delete it. Press ESC to recompile the script and close the Designer. If there are errors, the Designer will highlight them; if the Designer closes without problem, the recompile succeeded and the script should compile correctly in future.

See also "Using Domino Global WorkBench with Notes Release 4.6 databases" in Chapter 1.

Corrupted data action

Meaning: The internal structure of an action button has been corrupted.

Solution: Try opening and resaving the design element in the source database, and then update or rebuild the tagged and language databases. If that does not work you may have to re-create the design element from scratch.

Database "<database name>" already has a replica being synchronized (Set: "<setname>", Database: "<database name>").

Meaning: There are two databases on the server with the same replica ID. Synchronized databases should not be allowed to replicate together because the Language Synchronizer cannot distinguish between documents created by replication and genuine new documents.

Solution: Remove one of the database replicas. You may need to rebuild one or more of the language databases to make sure it has a different replica ID from any of the others in the synchronization set. When building, on the Create Language Database dialog box, choose 'Create database copy' as the Database Type.

Database property "Use JavaScript when generating pages" had to be turned on to support the switchbar

Meaning: The SwitchBar requires that this flag be set in order to function from the Web.

Solution: To stop this warning message from appearing in future projects, set the flag before you start localization. Go to the first tab of the Database infobox for the source database and check 'Web access: Use JavaScript when generating pages'.

Default language version used for design element "<element name>" instead of language "<language>"

Meaning: The WorkBench cannot find the required language version of a synchronization agent or dialog in DGWSTORE.NSF. If there is a version belonging to the same language family it will use that instead (for example it will use French-France instead of French-Canadian), otherwise it will use the reference language version.

Solution: Check that the language in question is present in DGWSTORE.NSF. If not, create it. See "Adding new languages" in Chapter 4.

Default navigator not found in icon note

Meaning: The default navigator cannot be found.

Solution: In Notes, right-click the source database icon to open its properties box. Tab to the 'Launch' panel. Select 'Open designated navigator in its own window' and then select the required default navigator in the 'Navigator' drop-down box.

Design element not present

Meaning: When trying to add synchronization features, the WorkBench could not locate a required Language Synchronizer design element in DGWSTORE.NSF.

Solution: Try reinstalling Domino Global WorkBench to re-create DGWSTORE.NSF. If you have any custom languages in your glossaries, you will have to re-create corresponding entries in the new DGWSTORE.NSF. See "Adding new languages" in Chapter 4.

Duplicates are not allowed

Meaning: You are trying to synchronize databases that have the same replica ID. Synchronized databases should not be allowed to replicate together because the Language Synchronizer cannot distinguish between documents created by replication and genuine new documents.

Solution: Remove one of the database replicas. You may need to rebuild one or more of the language databases to make sure it has a different replica ID from any of the others in the synchronization set. When building, on the Create Language Database dialog box, choose 'Create database copy' as the Database Type.

Empty buffer to translate

Meaning: The internal parser object has received an empty buffer to translate.

Solution: You can ignore this message as no data corruption or loss results.

Error building this language

Meaning: One or more errors were encountered when building this language database.

Solution: This is a summary message. It will be preceded by more specific messages that pinpoint the problems.

Error converting HTML to BiDi

Meaning: When building a database for a right-to-left reading language, the HTML text could not be converted to the proper right-to-left display order.

Solution: You will have to fix the problem in the built language database.

Error loading a project

Meaning: An error was encountered loading the project. Possibly project information has been deleted from the project database (PROJECT.NSF) in the Domino Global WorkBench program directory.

Solution: Start the WorkBench, create a new project, and load the source and glossary databases into it. Recover the tagged database by choosing the File - Add Existing Tagged Database. Recover the language databases by choosing File - Add Existing Language Database. Before updating, re-tagging or rebuilding, check the tagging options (Actions - Set Tagging Options) and the build options (Actions - Set Building Options).

Error opening synchronization set

Meaning: An existing synchronization set document (in the Synchronizer configuration database) was found that corresponds to the set now being configured. The document could not be opened.

Solution: This is a summary message. It will usually be preceded by a more specific message that pinpoints the problem.

Error parsing rich text term

Meaning: The WorkBench encountered problem(s) processing rich text in a database.

Solution: This is a summary message. It will be preceded by more specific messages that pinpoint the problems.

Error saving synchronization set

Meaning: A synchronization set document (in the Synchronizer configuration database) could not be saved.

Solution: This is a summary message. It will usually be preceded by a more specific message that pinpoints the problem.

Error: Tag [tag] (Graphic Term) translation is a native image and cannot be built into a navigator

Meaning: The specified graphic term has been translated and then imported to a glossary as a native image (.GIF or .JPEG). The use of native images is not supported in navigators, and so the image cannot be built to the language database.

Solution: Delete the image from the translation document in the glossary, and then reimport it from a .BMP file.

Error: Tag not found

Meaning: When creating a language database, The Workbench could not find a tagged term (from the tagged database) in any of the loaded glossaries. Either the term has been deleted from the glossary or the necessary glossary was not loaded.

Solution: Make sure the appropriate glossaries are loaded. If necessary, use the WorkBench to rebuild the tagged database and then rebuild the language database.

Error updating synchronization set

Meaning: An existing synchronization set document (in the Synchronizer configuration database) was found that corresponds to the current set, but it could not be updated.

Solution: This is a summary message. It will usually be preceded by a more specific message that pinpoints the problem.

Failed to load the icon

Errors encountered while processing the database icon.

Solution: Try opening and resaving the database icon in the source database (choose View - Design - Other), and then update or rebuild the tagged and language databases. If that does not work you may have to re-create the icon from scratch.

Failed to retrieve icon info

Errors encountered while processing the database icon.

Solution: Try opening and resaving the database icon in the source database (choose View - Design - Other), and then update or rebuild the tagged and language databases. If that does not work you may have to re-create the icon from scratch.

Icon bitmap not found in icon note

Errors encountered while processing the database icon.

Solution: Try opening and resaving the database icon in the source database (choose View - Design - Other), and then update or rebuild the tagged and language databases. If that does not work you may have to re-create the icon from scratch.

Icon note and database title

Errors encountered while processing the database icon.

Solution: Try opening and resaving the database icon in the source database (choose View - Design - Other), and then update or rebuild the tagged and language databases. If that does not work you may have to re-create the icon from scratch.

Incorrect signature found

Meaning: The internal structure of this design element has been corrupted.

Solution: Try opening and resaving the design element in the source database, and then update or rebuild the tagged and language databases. If that does not work you may have to re-create the design element from scratch.

Internal error BBLParse

Meaning: The WorkBench encountered problem(s) processing a database.

Solution: This is a summary message. It will be preceded by more specific messages that pinpoint the problems.

Invalid document structure (missing hotspot end)

Meaning: One or more hotspots in this design element have been corrupted.

Solution: Try opening and resaving the design element in the source database, and then update or rebuild the tagged and language databases. If that does not work you may have to re-create the design element from scratch.

Length restriction - text exceeded allowed limit (n characters) the text has been truncated

Meaning: The text has been reduced to enable it to fit within the maximum limit allowed by Notes. If you see this message when creating a tagged database you can usually ignore it. However, if you see the message when building language databases, you will have to take action because it means that the specified text in your language database has been truncated.

Solution: Shorten the translation that is causing the problem by editing the appropriate language document in the glossary.

Lotus component DLL error. Exiting application!

Meaning: One of the shared components used by the WorkBench could not be located (for example SmartIcons, StatusBar, or InfoBox).

Solution: Reinstall Domino Global WorkBench.

LotusScript line length exceeds 2000 character limit.

Meaning: A line of LotusScript exceeds the specified limit. Domino Global WorkBench copies the LotusScript unchanged to the destination database.

Solution: Check that an %INCLUDE statement is not causing a LotusScript line to exceed the permitted maximum.

LotusScript string missing end-quote.

Meaning: A line of LotusScript is missing its end quote character. Domino Global WorkBench assumes an end quote at the end of the line.

Solution: You can usually ignore this warning. If the end quote is missing from the end of the line, you do not need to do anything. Otherwise, insert the end quote in its correct position.

Multilingual databases cannot be Synchronized with unilingual databases. Please refer to the online documentation.

Meaning: You have chosen Actions - Synchronize Language Databases or clicked Synchronize Languages with both a multilingual database and a unilingual database selected. Language synchronization can work either in a set of unilingual database or in a single multilingual database, not both.

Solution: Select a group of unilingual databases or a multilingual one and try again.

No default language selected for this language database

Meaning: No default language has been set in this multilingual language database. With Release 5 of the WorkBench it is not possible to define a multilingual database without a default language, so you should not see this message.

Solution: Define and build the multilingual database again from scratch.

No default language set

Meaning: No default language has been set in this multilingual language database. With Release 5 of the WorkBench it is not possible to define a multilingual database without a default language, so you should not see this message.

Solution: Define and build the multilingual database again from scratch.

No field found in this design element!

Meaning: No field has been found in the source design element. The design element or the database is corrupt.

Solution: Check the design element in your source database.

No source database selected

Meaning: A process has been launched and the WorkBench couldn't find the selected source database. Perhaps the source database is on a network drive that has become unavailable.

Solution: Check that the source database is available and try again.

No synchronizer form added to : <name>"

Meaning: The WorkBench encountered a form that needed synchronizing, but which was not selected for language synchronization.

Solution: Select the form for language synchronization and choose Actions - Synchronize Language Databases again.

Not a valid glossary:

Meaning: The glossary may be an old format (before 5.0), not a glossary database at all, or corrupt in some other way.

Solution: If the glossary is a Notes Global Designer Release 4.6 glossary, you must upgrade it before you can use it with Domino Global WorkBench Release 5. See "Upgrading an existing glossary" in Chapter 4.

Not a valid Reports Database

Meaning: A reports database must be based on the template DGWREP.NTF.

Solution: Enter the name of a valid reports database.

Notes failed to start. Check that the Notes directory is on your path.

Meaning: Notes did not execute when requested to do so by the WorkBench.

Solution: Check that Notes is installed. Make sure that your PATH statement includes the Lotus Notes program directory. There should be a SET PATH command in your AUTOEXEC.BAT file which includes this directory.

One multilingual database or multiple unilingual databases must be selected for Synchronization

Meaning: You have chosen Actions - Synchronize Language Databases or clicked Synchronize Languages, but with either no language database selected or a single unilingual database selected.

Solution: Synchronization can only work when at least two languages are supported. You must select one multilingual database or two or more unilingual ones.

Out of memory

Meaning: Your computer is low on memory.

Solution: Close one or more applications and try again.

Reduce the length of the title before rebuilding it

Meaning: The title of this design element has been reduced to enable it to fit within the maximum limit allowed by Notes. If you see this message when creating a tagged database you can usually ignore it. However, if you see the message when building language databases, you will have to take action because it means that the specified text in your language database has been truncated.

Solution: Shorten the translation that is causing the problem by editing the appropriate language document in the glossary.

Registry error while accessing the key <name>

Meaning: WorkBench encountered problems reading or writing to the Windows registry.

Solution: If you are using Windows NT you may not have the security access you need. You should have Administrator access when running the WorkBench.

Some aliases in the title will be removed by the WorkBench

Meaning: For the WorkBench to modify the title of this element it was necessary to remove an alias. When the WorkBench removes aliases it removes the rightmost alias first. Removing an alias may affect the functionality of the database.

Solution: Either shorten the title-plus-alias combination in the source database, and edit any formulas or script that the change may affect, or reinstate the missing alias (which should not be translated) manually in the individual language databases.

Sorry, an uncorrectable error has occurred. Press ENTER to abort the application.

Meaning: The WorkBench has encountered a fatal error.

Solution: The only course of action is to press ENTER to stop the WorkBench. You should then be able to restart it. If you have difficulty restarting the WorkBench, you should restart your computer. Please report the problem to Lotus technical support.

Structure of view is incorrect

Meaning: The internal structure of a view has been corrupted.

Solution: Try opening and resaving the view in the source database, and then update or rebuild the tagged and language databases. If that does not work you may have to re-create the view from scratch.

Syntax/formatting problem (e.g. comma not allowed in keyword translation)

Meaning: A syntax error has been encountered when building a language database.

Solution: The most likely cause of this error is that some translated text includes one or more illegal characters. A keyword, for example, must not contain a comma. Correct the translation that is causing the problem by editing the appropriate language document in the glossary.

Tag not found in any loaded glossary

Meaning: When creating a language database, the WorkBench could not find a tagged term (from the tagged database) in any of the loaded glossaries. Either the term has been deleted from the glossary or the necessary glossary was not loaded.

Solution: Make sure the appropriate glossaries are loaded. If necessary, use the WorkBench to rebuild the tagged database and then rebuild the language database.

Term not found in any loaded glossary

Meaning: You are likely to see this message when you create a tagged database with 'Never Prompt' selected and 'Create New Terms' unchecked (on the Prompt panel of the Tagging Options dialog box). It indicates that a piece of text in the source database is not present in any of the loaded glossaries.

Solution: Re-create the tagged database with different tagging options so that the missing term can be created in the default glossary.

Term not found tag/term

See the previous message.

TermManager creation error

Meaning: Error creating a new term in the glossary.

Solution: Your computer may be low on memory: close some applications and then try rebuilding the tagged database.

Text with this style will not be tagged

Meaning: The text referred to had the style DO_NOT_TAG set and will therefore not be tagged.

Solution: You should not need to take any action: text marked with this paragraph style should not be translated.

The data in this document is corrupted

Meaning: The internal structure of the document has been corrupted.

Solution: Try opening and resaving the document in the source database, and then update or rebuild the tagged and language databases. If that does not work you may have to re-create the document from scratch.

The "Generate HTML for all fields" property was turned on to support the SwitchBar in the form : <formname>

Meaning: The SwitchBar requires that this flag be set in order to function from the Web.

Solution: To stop this warning message from appearing in future projects, set the flag before you start localization. Go to the second tab of the Form properties infobox for each form in the source database and check 'Generate HTML for all fields'.

The glossary database <glossary name> is already loaded

Meaning: The glossary has already been added to this project.

Solution: Load another glossary if you need to.

The specified database <database name> already exists. Overwrite?

Meaning: You are creating a tagged or language database for the first time, and the name you supplied for the new database is that of an existing database.

Solution: Click OK to overwrite the existing file or Cancel to go back and supply a different file name.

The WorkBench couldn't add its internal tag to the <design element> title without truncating it because of a Notes length restriction:

Meaning: The text has been truncated in the tagged database to enable it to fit within the maximum limit allowed by Notes.

Solution: You can usually ignore this message. It has no effect on your source database, and provided the translations for the specified title are within Notes limits, the title will build correctly in the language databases.

The WorkBench will truncate the <design element> title of this design element because of a Notes length restriction:

Meaning: The text has been reduced to enable it to fit within the maximum limit allowed by Notes. If you see this message when creating a tagged database you can usually ignore it. However, if you see the message when building language databases, you will have to take action because it means that the specified text in your language database has been truncated.

Solution: Shorten the translation that is causing the problem by editing the appropriate language document in the glossary.

This design element is a Web Navigator and cannot be selected for processing in DGW. Remove the Web Browser compatible flag in Notes before selecting it.

Meaning: The WorkBench cannot tag navigators marked 'Web browser compatible'.

Solution: See Web navigators in Chapter 6.

Translating database title:

Meaning: This is an information message that you may see when building language databases.

Solution: None needed.

Translation missing (language) - tag, term

Meaning: When building a language database, a translation for the specified term could not be found. The text will appear in the language of the source database.

Solution: Provide a translation for the term in the appropriate glossary and rebuild the language database.

Unable to add language definition information to profile document

Meaning: For every language that you intend to synchronize there must be a language definition document in DGWSTORE.NSF (in the Domino Global WorkBench program directory). This document is either missing, or it does not match the language definition in the glossary.

Solution: Open DGWSTORE.NSF in Notes and check that the appropriate language definition document exists. If the document does not exist, create it. If it does exist, check that its language ID matches the language ID for the language as defined in the glossary. Note that the language ID in the glossary has no hyphen (for example 'ENUS') whereas in DGWSTORE.NSF it does have a hyphen (for example, 'en-US').

Unable to add synchronization information

Meaning: Error adding synchronization information.

Solution: This is a summary message. It will be preceded by more specific messages that pinpoint the problems.

Unable to add synchronization subform to form : <formname>

Meaning: The required synchronization subform cannot be appended to the specified form.

Solution: This is a summary message. It will be preceded by more specific messages that pinpoint the problem.

Unable to add the database <file name> to the project. It is not a DGW language database.

Meaning: You have tried to add a language database to the project using File - Add Existing Language Database, but the file you specified is not a Domino Global WorkBench database.

Solution: Check that the file you specified is the right one. If you cannot add the file you will have to rebuild it.

Unable to add the file <file name>. This is not a valid glossary. The glossary will be removed from the project. Would you like to proceed?

Meaning: The WorkBench does not recognize the specified file as a glossary.

Solution: If you click Yes the WorkBench will remove it from the project. If you click No, it will not be removed from the glossary. You can then open it in Notes and try to resolve the problem.

Unable to add the language database <file name>. The language database will be removed from the project. Would you like to proceed?

Meaning: A language database referenced in the project is no longer available. The most likely cause is that it has been deleted or moved, or is on a network drive that is not currently accessible.

Solution: If you click Yes the WorkBench will remove it from the project. If the database has not been deleted, click No and try again later when it is available.

Unable to add the source database <name>.

Meaning: A source database referenced in the project is no longer available. The WorkBench will automatically remove it from the project.

Solution: The most likely cause is that it has been deleted or moved, or is on a network drive that is not currently accessible. If the database has not been deleted, you can add it to the project again by choosing File - Add Source Database. Then proceed as explained below under "Unable to load the project:" to recover the other databases associated with it.

Unable to add the tagged database <name>. The tagged database will be removed from the project. Would you like to proceed?

Meaning: A tagged database referenced in the project is no longer available. The most likely cause is that it has been deleted or moved, or is on a network drive that is not currently accessible.

Solution: If you click Yes the WorkBench will remove it from the project. If the database has not been deleted, click No and try again later when it is available.

Unable to compile a formula after processing it:

Meaning: An error was encountered while attempting to recompile a formula.

Solution: The most likely cause of this error is that some translated text in the script or formula has made its syntax invalid. Check in the language database that the translation does not include any special characters that are used in formulas or script, for example double quotes (") or semicolons (;). Check that no program keywords have been translated. Correct the translation that is causing the problem by editing the appropriate language document in the glossary.

Unable to create a new glossary entry for the term:

Meaning: The WorkBench was unable to create the term in the glossary database.

Solution: Check that the glossary is not read only, that you have the necessary rights to it and that there is space available on your hard disk. Open the glossary in Notes to check that the file has not been corrupted. If the glossary is held on a CD it will usually be read only: you must copy it to a hard disk before the WorkBench can use it. You may want to create a new glossary to receive new terms (specify it on the Basics panel of the Tagging Options dialog box).

Unable to create database definition in synchronization set

Meaning: Every database in a synchronization set has an information document in the synchronizer configuration database. The WorkBench was unable to create one or more of these documents.

Solution: This is a summary message. It will usually be preceded by a more specific message that pinpoints the problem.

Unable to create synchronization set

Meaning: Error creating a synchronization set document in the synchronizer configuration database.

Solution: This is a summary message. It will usually be preceded by a more specific message that pinpoints the problem.

Unable to create the specified font

Meaning: The font specified could not be created.

Solution: The most likely reason for this error is lack of memory. Close some applications and try the operation again.

Unable to find a language definition for %s in DGWStore.NSF

Meaning: For every language that you intend to synchronize there must be a language definition document in DGWSTORE.NSF (in the Domino Global WorkBench program directory). This document is either missing, or it does not match the language definition in the glossary.

Solution: Open DGWSTORE.NSF in Notes and check that the appropriate language definition document exists. If the document does not exist, create it. If it does exist, check that its language ID matches the language ID for the language as defined in the glossary. Note that the language ID in the glossary has no hyphen (for example 'ENUS') whereas in DGWSTORE.NSF it does have a hyphen (for example, 'en-US').

Unable to find the Notes run-time DLL's. Program aborted!

Meaning: Notes program files could not be found.

Solution: Check that Notes is installed. Make sure that your PATH statement includes the Lotus Notes program directory. There should be a SET PATH command in your AUTOEXEC.BAT file which includes this directory.

Unable to get the short tag associated with a term

Meaning: The WorkBench is trying to tag to a glossary entry, but cannot. You may see this message if, for some reason, the WorkBench fails to create a term in a glossary and then tries to tag to the non-existent new term.

Solution: Check that you have at least Author access rights in the glossary's access control list (ACL). Check that the disk holding the glossary is not full.

Unable to get the Update information for the tagged database

Meaning: The update information for this database could not be retrieved.

Solution: This is a summary message. It will be preceded by more specific messages that pinpoint the problems.

Unable to interpret Timedate value

Meaning: A time or date value has not been entered correctly.

Solution: Enter the data in a valid Notes format.

Unable to load the glossary <name>

Meaning: A glossary database referenced in the project is no longer available.

Solution: The most likely cause is that it has been deleted or moved, or is on a network drive that is not currently accessible. If the database has not been deleted, you can add it to the project again by choosing File - Add Glossary Database.

Unable to load the project:

Meaning: An error was encountered loading the project. Possibly project information has been deleted from the project database (PROJECT.NSF) in the Domino Global WorkBench program directory.

Solution: Start the WorkBench, create a new project, and load the source and glossary databases into it. Recover the tagged database by choosing the File - Add Existing Tagged Database. Recover the language databases by choosing File - Add Existing Language Database. Before updating, re-tagging or rebuilding, check the tagging options (Actions - Set Tagging Options) and the build options (Actions - Set Building Options).

Unable to load the project database: <name>

Meaning: The project database, PROJECT.NSF in the Domino Global WorkBench program directory, could not be loaded. You may not have the right access rights, or the project database may have been deleted or it may be corrupt.

Solution: Try opening PROJECT.NSF in Notes. Choose File - Database - Access Control and check that you have Designer access rights. If necessary, create a new project database (in the Domino Global WorkBench program directory) using the template DGWPROJ.NTF. To restore an existing project to the new project database, start the WorkBench, create a new project, and load the source and glossary databases into it. Recover the tagged database by choosing the File - Add Existing Tagged Database. Recover the language databases by choosing File - Add Existing Language Database. Before updating, re-tagging or rebuilding, check the tagging options (Actions - Set Tagging Options) and the build options (Actions - Set Building Options).

Unable to load the resource only DLL: <name>

Meaning: A file containing the resources required by the WorkBench could not be loaded. The file may have been deleted or the computer may be low on memory.

Solution: Close some applications and then try restarting the WorkBench. If that does not work, you should reinstall Domino Global WorkBench.

Unable to load the source database <name>. The source database will be removed from the project. Would you like to proceed?

Meaning: A source database referenced in the project is no longer available. The most likely cause is that it has been deleted or moved, or is on a network drive that is not currently accessible.

Solution: If you click Yes the WorkBench will remove it from the project. If the database has not been deleted, click No and try again later when it is available.

Unable to locate a bitmap resource <name>

Meaning: A bitmap resource could not be found when building. You are likely to see this message when you create a tagged database with 'Never Prompt' selected and 'Create New Terms' unchecked (on the Prompt panel of the Tagging Options dialog box). It indicates that a bitmap in the source database is not present in any of the loaded glossaries.

Solution: Re-create the tagged database with different tagging options so that the missing bitmap can be created in the default glossary.

Unable to open design element

Meaning: When trying to add synchronization features, the WorkBench could not open a required Language Synchronizer design element in DGWSTORE.NSF.

Solution: Try reinstalling Domino Global WorkBench to re-create DGWSTORE.NSF. If you have any custom languages in your glossaries, you will have to re-create corresponding entries in the new DGWSTORE.NSF. See "Adding new languages" in Chapter 4.

Unable to open Design view in DGWSTORE.NSF

Meaning: When trying to add synchronization features, the WorkBench could not open a design view in DGWSTORE.NSF.

Solution: Check that DGWSTORE.NSF is present in the Domino Global WorkBench program directory available database, and that you can open it. If you cannot access it try reinstalling Domino Global WorkBench.

Unable to open global subform from DGWStore.NSF

Meaning: When trying to add synchronization features, the WorkBench could not open the 'Global' subform in DGWSTORE.NSF.

Solution: Try reinstalling Domino Global WorkBench to re-create DGWSTORE.NSF. If you have any custom languages in your glossaries, you will have to re-create corresponding entries in the new DGWSTORE.NSF. See "Adding new languages" in Chapter 4.

Unable to open local subform from DGWStore.NSF

Meaning: When trying to add synchronization features, the WorkBench could not open the 'Local' subform in DGWSTORE.NSF.

Solution: Try reinstalling Domino Global WorkBench to re-create DGWSTORE.NSF. If you have any custom languages in your glossaries, you will have to re-create corresponding entries in the new DGWSTORE.NSF. See "Adding new languages" in Chapter 4.

Unable to open main view in synchronizer configuration database (SYNCCONF.NSF) on server "<servername>"

Meaning: The Synchronizer configuration database, SYNCCONF.NSF, is present in the data directory of the server, but a required view cannot be opened.

Solution: Check that you have sufficient rights in the database (the default is Designer).

Unable to open synchronization subform in DGWSTORE.NSF

Meaning: When trying to add synchronization features, the WorkBench could not access the subform in DGWSTORE.NSF.

Solution: Check that you can access DGWSTORE.NSF (in the Domino Global WorkBench program directory). Try reinstalling Domino Global WorkBench to re-create DGWSTORE.NSF. If you have any custom languages in your glossaries, you will have to re-create corresponding entries in the new DGWSTORE.NSF. See "Adding new languages" in Chapter 4.

Unable to open synchronizer configuration database (SYNCCONF.NSF) on server "<servername>"

Meaning: The synchronization configuration database, SYNCCONF.NSF, was not found in the data directory of the server to which you're attempting to deploy synchronized databases.

Solution: This is probably because the Language Synchronizer has not been installed on the server. See "Installing the Language Synchronizer and the Synchronizer configuration database" in Chapter 8.

Unable to open the project specified

Meaning: When trying to add synchronization features, the WorkBench could not open the current project in the project database, PROJECT.NSF, in the Domino Global WorkBench program directory. The project data may have been deleted or have become corrupt.

Solution: In Notes, try opening the project document in PROJECT.NSF. If the document is missing or corrupt, you will have to re-create it. To recover the project, start the WorkBench, create a new project, and load the source and glossary databases into it. Recover the tagged database by choosing the File - Add Existing Tagged Database. Recover the language databases by choosing File - Add Existing Language Database. Before updating, re-tagging or rebuilding, check the tagging options (Actions - Set Tagging Options) and the build options (Actions - Set Building Options).

Unable to open the language database

Meaning: During an update a language database referenced in the project could not be opened. You may not have the necessary access rights, or a disk or network error may have occurred.

Solution: Try opening the database in Notes. Choose File - Database - Access Control and check that you have at least Designer access rights. In the WorkBench choose View - Refresh (F9) and try the update again. If necessary, rebuild the database.

Unable to open the language database specified

Meaning: When trying to add synchronization features, the WorkBench could not open the source database.

Solution: The most likely cause is that it has been deleted or moved, or is on a network drive that is not currently accessible.

Unable to open the source note

Meaning: The design element in the source database could not be opened.

Solution: In the WorkBench, choose View - Refresh (F9) and try the procedure again. Try opening and resaving the design element in the source database. If that does not work you may have to re-create the design element from scratch.

Unable to open the source specified

Meaning: When trying to add synchronization features, the WorkBench could not open the source database.

Solution: The most likely cause is that it has been deleted or moved, or is on a network drive that is not currently accessible.

Unable to open the store database (DGWSTORE.NSF may be missing/misplaced)

Meaning: When trying to add synchronization features, the WorkBench could not open DGWSTORE.NSF, in the Domino Global Designer program directory.

Solution: The most likely cause is that it has been deleted or moved, or is on a network drive that is not currently accessible.

Unable to open the tagged database

Meaning: The tagged database could not be opened.

Solution: Try opening the database in Notes. Choose File - Database - Access Control and check that you have at least Editor access rights. In the WorkBench, choose View - Refresh (F9) and try the process again. If necessary, rebuild the database.

Unable to open translation subform from DGWStore.NSF

Meaning: When trying to add synchronization features, the WorkBench could not open the 'Translatable' subform in DGWSTORE.NSF.

Solution: Try reinstalling Domino Global WorkBench to re-create DGWSTORE.NSF. If you have any custom languages in your glossaries, you will have to re-create corresponding entries in the new DGWSTORE.NSF. See "Adding new languages" in Chapter 4.

Unable to perform the operation while an action is in progress

Meaning: You tried to carry out an operation that is forbidden while another operation is in progress. For example, you cannot exit the WorkBench while building.

Solution: Try again when the first operation has finished.

Unable to tag HTML due to unrecoverable syntax error. Copying HTML.

Meaning: A syntax error has been detected at the very end of the HTML buffer, and the tagging cannot be completed. In this case the WorkBench reverts to the original, untagged HTML.

Solution: Edit the HTML to resolve the syntax errors then retag/rebuild.

Unable to write profile document to language database

Meaning: The profile document in the language database specified cannot be written to.

Solution: This is a general error message: more detailed message(s) will pinpoint the problem.

Unable to write to language database

Meaning: When adding synchronization features, a language database could not be written to.

Solution: This is a general error message: more detailed message(s) will pinpoint the problem.

Unexpected end of HTML stream. Copying HTML.

Meaning: A piece of HTML text has not been terminated in a way that Domino Global WorkBench recognizes. The text will be copied, unchanged, to the destination database.

Solution: Check that sections of HTML have the correct ending tags.

Unexpected exception raised

Meaning: A general internal problem was encountered.

Solution: You will have to restart Domino Global WorkBench. If that does not work, try reinstalling. Please report the problem to Lotus technical support.

Unexpected token in HTML stream. Copying HTML. Token:

Meaning: A piece of HTML text includes a token that Domino Global WorkBench cannot process. The text will be copied, unchanged, to the destination database.

Solution: Check that the specified token is valid. Consider removing it.

Unsupported object

Meaning: An object has been encountered that is not supported by the WorkBench.

Solution: Please report the problem to Lotus technical support.

Unsupported record type in pass-thru HTML. Copying HTML.

Meaning: Pass-thru HTML has been encountered of a format that is not supported.

Solution: Make sure you are using the latest version of Domino Global WorkBench.

You cannot exclude the icon note for building!

Meaning: You have deselected the database's icon (under Resources ... Other) when building.

Solution: Select the icon and try again.

You cannot use update to make a single language database multilingual. You must rebuild!

Meaning: You have chosen Tagged Database - Update Language Database after adding a language to an existing language database.

Solution: You must rebuild the language database. Choose the database in the drop-down box below the tagged database icon and choose Tagged Database - Rebuild Language Database.

You must enter a report database name

Meaning: You have chosen to log WorkBench messages in a report database.

Solution: Enter the name of your report database.

You must specify a positive Synchronization schedule interval

Meaning: You have entered a zero or negative synchronization schedule interval on the Configure Synchronization Set dialog.

Solution: Enter a positive value.

You need either a multilingual database, or multiple unilingual databases created, in order to prepare them for synchronization

Meaning: You have chosen Actions - Synchronize Language Databases or clicked Synchronize Languages, but with either no language database selected or a single unilingual database selected.

Solution: Synchronization can only work when at least two languages are supported. You must select one multilingual database or two or more unilingual ones.

Glossary error and warning messages

The following is an alphabetical listing of the glossary error and warning messages. An explanation is given for each message and, where appropriate, suggested action.

Error messages

<Language> is already the reference language

Meaning: When changing the reference language of a glossary you have selected the language that is already the reference language.

Solution: To change to another reference language select a different active language.

An unselected language cannot be made reference language

Meaning: When changing the reference language of a glossary you have selected a language that is not an active language.

Solution: Select an active language, or make the currently selected language active, and retry.

Can't open selected database

Meaning: Before the Retrieve dialog box can open, you must select a glossary to retrieve terms from. The glossary you specified cannot be opened for some reason.

Solution: Try opening the glossary directly from Notes. If this is successful, close it and retry the retrieve operation. Otherwise refer to the database manager.

Each format parameter can only appear once in the formatting string

Meaning: In the Import dialog box, you have specified the same parameter more than once. The only parameter that is allowed to appear more than once is 'None'.

Solution: Check the parameters you have specified against your structured text file. If the same information appears in more than one column of the structured text file, change the parameter for all but one of the occurrences to 'None'.

File not present or empty

Meaning: In the Import dialog box, you must enter the name of the structured text file that holds the terms being imported.

Solution: Check that the file is correctly specified in the dialog box and that it contains the terms to be imported.

Language document already exists

Meaning: You have tried to add support for a language that is already supported.

Solution: Check the details of the language you are adding, and that it is not already supported by the glossary.

Language identifier already exists

Meaning: When adding support for a new language in a glossary, you have entered a language identifier that is already in use.

Solution: Enter a different language identifier (check that the language you are adding is not already supported by the glossary).

Not all translation placeholders are translated for this language

Meaning: You cannot change the reference language of a glossary unless all the translation documents for the new reference language are marked either 'Translated' or 'Approved'.

Solution: Go to the 'Untranslated terms' view in the glossary to find the untranslated documents in the proposed reference language. Translate them and retry.

Please select a glossary

Meaning: Before the Retrieve dialog box can open, you must select a Domino Global WorkBench Release 5 glossary from which to retrieve terms.

Solution: Select a valid glossary.

Term has been marked as a DNT so it can not be edited

Meaning: The term has been marked Prevent Translation (do not translate, or DNT). The translations for a term marked in this way cannot be edited.

Solution: If you really do need to edit the document, select the term document (not a translation document) and click Allow Translation. You will now be able to edit its translations.

The delimiters must be different

Meaning: In the Import dialog box, you must specify one delimiter to separate the main fields in your structured text file and another to separate term types, if they are present in your structured text file. The delimiters must be different.

Solution: Check the delimiters you have entered to make sure they are different. If necessary, change the delimiters in the structured text file.

The reference languages do not match. Please choose another glossary.

Meaning: Before the Retrieve dialog box can open, you must select a glossary from which to retrieve terms. The retrieving of translations from one glossary into another works by matching the reference language text of terms in the two glossaries. For this to be successful the reference languages of the two glossaries must be the same.

Solution: Retrieve translations from another glossary or retrieve these translations into another glossary.

The reference term must be included in the formatted input string

Meaning: In the Import dialog box, you have not specified reference term (the text of the term in the glossary's reference language) as one of the parameters. You cannot import from a structured text file that does not include this as one of its elements.

Solution: Make sure that your structured text file includes the reference language text and that it is specified as one of the parameters.

The reference translation is not editable

Meaning: You cannot edit the reference language translation document.

Solution: Edit the main term document instead. Changes you make will be copied to the translation document.

The reference translation is not editable. To make changes edit the reference term

Meaning: You cannot edit the reference language translation document.

Solution: Edit the main term document instead. Changes you make will be copied to the translation document.

The RTF_Term field must contain some text

Meaning: When creating a new rich text term in a glossary you tried to save without entering text for the term.

Solution: Enter text for the term and retry.

The selected database is not a DGW 5.0 glossary. Please select another

Meaning: Before the Retrieve dialog box can open, you must select a Domino Global WorkBench Release 5 glossary from which to retrieve terms.

Solution: Select a valid glossary. If the glossary you are trying to access comes from Notes Global Designer Release 4.6, you will have to upgrade it first.

There are no languages selected in this database

Meaning: There are no active languages in the glossary. The language of the terms you are importing must be one of the active languages in the glossary.

Solution: In the glossary, choose the Available Languages view, select the language, and click Select.

There are no more contexts in the glossary

Meaning: When using the Translate/Review dialog box you have tried to select a design element beyond the last one in the glossary.

Solution: No action necessary.

There are no more terms in this context. Would you like to go to the next context?

Meaning: When using the Translate/Review dialog box you have tried to navigate beyond the last piece of text in the current design element.

Solution: Click Yes to go to the next design element or No to stay with the current one.

There are no previous contexts in the glossary

Meaning: When using the Translate/Review dialog box you have tried to select a design element before the first one in the glossary.

Solution: No action necessary.

There are no previous terms in this context. Would you like to go to the preceding context?

Meaning: When using the Translate/Review dialog box you have tried to navigate back from the first piece of text in the current design element.

Solution: Click Yes to go to the previous design element or No to stay with the current one.

There are no term documents selected for translation prevention

Meaning: You are trying to mark a term document Prevent Translation (DNT) without first selecting one.

Solution: Select the term document or documents for which you want to prevent translation and click Prevent Translation.

There are no term documents selected to allow for translation

Meaning: You are trying to mark a term document 'Allow translation' without first selecting one.

Solution: Select the term document or documents for which you want to prevent translation and click Allow Translation.

There are no term documents selected to delete

Meaning: You are trying to delete a term document without first selecting one.

Solution: Select the term document or documents you want to delete and click Delete Terms. If, after deleting a term, you want to make the WorkBench re-tag to a new term without redoing the whole tagging process, you can force an update. See "Forcing an update" in Chapter 6.

There are no term documents selected to mark as current

Meaning: You are trying to mark an obsolete term document as current without first selecting one.

Solution: Select the obsolete term document or documents you want to mark current and click Mark Current.

There are no term documents selected to mark as obsolete

Meaning: You are trying to mark a term document obsolete without first selecting one.

Solution: Select the term document or documents you want to mark obsolete and click Mark Obsolete.

There are no terms to process in the current glossary

Meaning: You are trying to retrieve translations into an empty glossary.

Solution: Make sure that the host glossary (the one you are retrieving into) contains terms in the reference language and that the language you want to retrieve is an active language. Retrieval works by matching the reference language text of terms in the host glossary against the reference language text of terms in the glossary you are retrieving from. Only when there is a match can retrieval happen.

There are no unprocessed pseudo terms in this glossary or you have not selected any pseudo language documents

Meaning: Terms that have been pseudo-translated cannot be pseudo-translated a second time. If you try to do so you will see this message.

Solution: Select documents that have not already been pseudo-translated and retry.

There is no current term to work from (database may be empty)

Meaning: Before the Translate/Review dialog box can open there must be at least one translation document in the glossary database.

Solution: Try again when the glossary contains documents to translate.

This is not a language document

Meaning: In order to make a language active or inactive you must first select a language document.

Solution: In the glossary, choose the Available Languages view, select a language document and click Select or Deselect.

This language is already deselected

Meaning: You cannot make a language inactive if it is already inactive.

Solution: In the glossary, choose the Available Languages view, select a different language document and click Deselect.

This language is already selected

Meaning: You cannot make a language active if it is already active.

Solution: In the glossary, choose the Available Languages view, select a different language document and click Select.

Translation has been marked as a DNT so it can not be edited

Meaning: The term has been marked Prevent Translation (do not translate, or DNT). The translations for a term marked in this way cannot be edited.

Solution: If you really do need to edit the document, select the term document (not a translation document) and click Allow Translation. You will now be able to edit its translations.

View '(MultiPurpose)' missing from glossary

Meaning: The specified hidden view could not be found while retrieving translations from another database.

Solution: Ask the manager of the glossary database to fetch the necessary view from the glossary template by refreshing the database design (choose File - Database - Refresh Design).

You can only select one language to be the reference language

Meaning: When changing the reference language of a glossary you have selected more than one active language.

Solution: Select a single active language, the one you want to make the reference language, and try again.

You cannot edit a reference object term

Meaning: You cannot edit objects such as bitmaps in a glossary.

Solution: See "Translating graphics" in Chapter 5 for information about translating bitmaps.

You do not have Access Rights to Setup/Update this glossary. Please contact the Database Manager

Meaning: You need Manager rights in the access control list (ACL) in order to upgrade a glossary.

Solution: Ask the database manager to give you manager rights, or to carry out the upgrade operation herself.

You do not have sufficient rights to create terms. No new terms will be created

Meaning: You need author rights in order to create a new term in a glossary.

Solution: Ask the database manager to give you author rights.

You do not have sufficient rights to perform this operation

Meaning: You need to have additional rights assigned in the glossary's access control list (ACL) in order to carry out the operation.

Solution: Ask the database manager to give you the rights you need, or to carry out the operation herself.

You first need to select the Pseudo language in the Available Languages view

Meaning: In order to carry out a pseudo-translation, the 'language' Pseudo must be active in the glossary.

Solution: In the glossary, choose the Available Languages view, select Pseudo, and click Select.

You must choose a new reference language before you can deselect the current reference language

Meaning: You cannot make a language inactive if it is the reference language.

Solution: Check that you really want to make the reference language inactive. If you do, choose the Available Languages view, select a different language to be the reference language and click Select as Reference. Then select this document again and click Deselect.

You must choose a reference language

Meaning: When creating a new glossary, or upgrading one, you must choose a reference language. This is usually the language of the source databases you are localizing.

Solution: Select a reference language and retry.

You must enter a numeric percentage amount for the expansion of the pseudo-translation

Meaning: In the Pseudo-translate dialog box, you must enter a number in the 'Expand by' box. If you choose to pseudo-translate by expanding text, it will be expanded by this percent.

Solution: Enter a number in the 'Expand by' box.

You must enter term text - textual rich text and plain text terms can only be created this way

Meaning: When creating a new term in a glossary you tried to save without entering text for the term (either plain text or rich text).

Solution: Enter text for the term and retry.

You must have deletion rights to the glossary to perform this operation

Meaning: The access control list (ACL) for the database does not give you the right to delete documents. You need this, for example, in order to make a language inactive (all translations in that language are then deleted).

Solution: Check that you really want to delete the documents or make the language inactive. If you do, ask the database manager to give you deletion rights, or to carry out the operation herself.

You must input a filename

Meaning: In the Export dialog box, you must enter the name of the structured text file that holds the terms you are importing.

Solution: Check that you have specified the file correctly and retry.

You must input a valid filename

Meaning: In the Import dialog box, you must enter the name of the structured text file that holds the terms being imported.

Solution: Enter a valid file name and retry.

You must select a document

Meaning: You have tried to carry out a glossary operation that requires a document to be selected, but without making any such selection.

Solution: Select the document(s) you want to process and retry.

You must select a language to export

Meaning: In the Export dialog box, you must select the language to be exported.

Solution: Select the appropriate language and retry.

You must select a language to import

Meaning: In the Import dialog box, you must specify the language of the terms you are importing.

Solution: Select a language in the drop-down box and retry. You can only import a language that is supported by the glossary.

You must select a source language

Meaning: In the Export dialog box, when exporting a language pair, you have not selected the first language of the pair.

Solution: Select the appropriate language and retry.

You must select one or more languages

Meaning: In the Retrieve dialog box, you must specify the language(s) of the translations you want to retrieve.

Solution: Select one or more languages in the 'Select Language(s)' list.

You must specify a delimiter string

Meaning: In the Import dialog box, you must specify the delimiter that is used to separate the main fields in your structured text file.

Solution: Enter the delimiter (often a comma or a semicolon) and retry.

You must specify a term type delimiter string

Meaning: In the Import dialog box, you must specify the delimiter that is used to separate term types, if they are present in your structured text file.

Solution: Enter the delimiter (often a colon) and retry.

You must type one single character to be the pseudo character

Meaning: In the Pseudo-translate dialog box, you must enter only one character for use when expanding text.

Solution: Enter one character in the 'Using character' box.

You need Manager Rights to update this glossary. Please contact the Database Manager

Meaning: You need manager rights in order to upgrade a glossary.

Solution: Ask the database manager to give you manager rights, or to carry out the upgrade operation herself.

Warning messages

The following table lists some of the warning messages you may encounter when working in a glossary.

Message	Meaning
Deleting term duplicates may result in missing information in a built language database.	Although duplicate terms may seem unnecessary in a glossary, it is possible that all or some of them are used in one tagged database or another. If you do delete duplicates watch out for reports of missing text when you build language databases. If, after deleting a term, you want to make the WorkBench re-tag to a new term without redoing the whole tagging process, you can force an update.
DGW: The comment for this term is too large to display. Refer to the translation document or reference term directly.	Access the term or translation document from one of the Notes views.
Language documents can't be cut or deleted.	You cannot cut or delete translation documents except by deleting their parent term document. When you delete a term document, all its translation documents are deleted as well.
Mismatched quotes found on line:	When importing from a structured text file, opening quotes were found without matching closing quotes. Check the structured text file.
Modifying term text will result in obsolete translations. Translation text will be restored to the reference term text.	If you change the text in a main term document, the text in all its translation documents will be changed to match, and they will be marked Untranslated.
Not all terms and translations can be used with Translate/Review.	There are graphic and/or rich text terms that need translation. These cannot be handled by the Translate/Review dialog.
Note: Rich text terms cannot be imported back into the glossary.	This is a warning that rich text terms cannot be imported.
Selected terms and their translations will now be deleted, selected translation documents with no selected term will be ignored.	When deleting terms, it is the selection of main term documents that counts; selection of translation documents has no effect. Selected term documents, and their translation documents, are deleted. Translation documents in the selected list are ignored.
The current term has not been saved, do you wish to save it?	If you click Yes, the term will be saved and marked as Translated.
The database information document can't be cut or deleted.	You cannot cut or delete the Database Information document.
The translations of all selected term documents, will be marked as untranslated and editable.	You will see this message when you select one or more term documents and click Allow Translation.
The translations of all selected term documents will be marked as translated and non editable. Translation text will be restored to the reference term text.	You will see this message when you select one or more term documents and click Prevent Translation.
There are more Terms in this category than can be loaded together. Use the arrow buttons to navigate.	Arrow buttons appear at the top and bottom of the list box if there are many pieces of text to process. Clicking an arrow button loads the next or previous group of terms.
There are no terms in this category suitable for translation with the Reviewer/Translator.	There are no translatable terms in this design element, or only graphic and/or rich text terms, which cannot be handled by the Translate/Review dialog.
There is context information missing for this term. The Reviewer/Translator needs this information.	The Translate/Review dialog uses context information in order to navigate from term to term. Access the term document from one of the Notes views and add the missing context information.

Message	Meaning
This glossary has not been fully updated to version 5.0.5. This operation may take some time.	Time warning when upgrading a glossary from Notes Global Designer 4.6.
This term has not been marked translated. Do you wish to do so?	If you click Yes, the term will be marked as Translated.
This will remove all documents in your database that have no parents, and may take some time.	A time warning. You see this message when you delete orphan documents. Orphans are translation documents that no longer have associated main term documents. These can arise if a term is deleted from a glossary replica that contains only a subset of the supported languages. Translation documents for the languages not in the subset replica will then remain in one or more of the other replica(s).
This will remove all Translation Replication/Save Conflicts of a Non-Linguistic nature in your database and may take some time.	A time warning. Sometimes a replication conflict can result in two translation documents for the same term and the same language. In cases where one has status Translated or Approved and the other has status Untranslated, you can delete the latter automatically by clicking Delete Non-Linguistic Conflicts.
Translation documents will now be created. This may take some time.	Time warning when you make a new language active.
Translation documents will now be deleted for all deselected languages. This may take some time.	Time warning when you make a new language inactive.
Use the Delete Terms action to delete terms and translations.	You cannot cut or delete a term document by pressing the Delete key. You must use the Delete Term action. When you delete a term document, all its translation documents are deleted as well.

Synchronization error and warning messages

This is an alphabetical listing of synchronization error and warning messages.

<User> does not have Manager access rights to the database: <dbname>

Meaning: The user specified (the last person to edit the synchronization set) does not have manager access to the database specified.

Solution: Either give the specified person manager access to the database or have an authorized manager edit the synchronization set document, thus becoming its 'owner'.

A non-original-language document was found unexpectedly

Meaning: Database internal structure error.

Solution: Check that the database has been built with the latest version of Domino Global WorkBench.

Additional database since last synchronization. Processing all documents.

Warning: An additional database has appeared since the last synchronization, so all documents are being synchronized from scratch (normally only modified documents are synchronized).

Cannot load multiple copies of the nsync task

Meaning: You are trying to load a second copy of the server task.

Solution: Only a single copy of the Language Synchronizer task may be loaded on a server.

Cannot open synchronization configuration database: <name>

Meaning: The synchronization configuration database cannot be opened.

Solution: Check that the database, SYNCCONF.NSF is present in the server's data directory and is not locked by another process. Try opening the database directly in Notes.

Cannot synchronize just one unilingual database.

Meaning: A synchronization set can comprise two or more unilingual databases or one multilingual database.

Solution: Check that your synchronization set document specifies two or more unilingual databases or one multilingual database.

Converting a user copy of a synchronized translation to local

Warning: A translation document (a document created by the Language Synchronizer) has been copied manually. This new document is being converted to 'local' synchronization type so that it won't interfere with synchronization.

Database design invalid for synchronizing. Synchronizer features may not have been added to this database.

Meaning: Database internal structure error.

Solution: Check that the database has been built with the latest version of Domino Global WorkBench.

Database design modified since last synchronization. Processing all documents.

Warning: A modification to the design of one or more of the database has been encountered since last synchronization, so all documents are being synchronized from scratch (normally only modified documents are synchronized).

Document found with an unsupported language: <language>

Warning: The language identifier found in a document is not supported by the database in which it is contained. This translation document is being deleted.

Document signatures are inconsistent for the following job: <setname>

Meaning: The documents for the synchronization set described (the set document and its responses, in the synchronization configuration database) are not all signed by the same author, or else a signature is corrupt.

Solution: Try editing and saving any of the documents.

Error fixing up a response document.

Meaning: A translation document created from a response document cannot be made a response to the language version of the corresponding main document.

Solution: Further synchronization may correct this. Otherwise you may see an orphan document.

Error generating statistics on this synchronization set

Warning: The Language Synchronizer could not update the statistics document for this synchronization set.

Error opening database for synchronization:

Meaning: The database as defined in the synchronization set cannot be opened.

Solution: This is a general error message: more detailed message(s) will pinpoint the problem.

Error opening matching document:

Meaning: A translation document (a document created by the Language Synchronizer) cannot be opened. Synchronization will continue.

Solution: This is a general error message: more detailed message(s) will pinpoint the problem.

Error opening profile document:

Meaning: A profile document, which contains synchronization settings, cannot be opened in the specified database.

Solution: Check that the database has been built with the latest version of Domino Global WorkBench. Rebuild if necessary.

Error opening source document:

Meaning: A source document (an original 'translatable' or 'global' document) cannot be opened. Synchronization will continue.

Solution: This is a general error message: more detailed message(s) will pinpoint the problem.

Error reading the configuration database

Meaning: The synchronization configuration database cannot be read.

Solution: Try opening the database, SYNCCONF.NSF directly in Notes. Check that it is the same version level as the Language Synchronizer server task.

Error retrieving matching documents

Meaning: A matching document found could not be opened. Possibly a sharing violation.

Solution: This should correct itself the next time the Language Synchronizer runs.

Error synchronizing documents:

Meaning: An error was encountered synchronizing documents.

Solution: This is a general error message: more detailed message(s) will pinpoint the problem.

Error with documents being synchronized

Meaning: An error was encountered synchronizing documents.

Solution: This is a general error message: more detailed message(s) will pinpoint the problem.

Invalid synchronization document format

Meaning: There is an internal structure error in the document.

Solution: Check that the database has been built with the latest version of Domino Global WorkBench. Try re-creating the document.

Invalid Synchronizer profile document. Synchronizer features may not have been added to this database.

Meaning: Internal database structure error.

Solution: Check that the database has been built with the latest version of Domino Global WorkBench. Rebuild if necessary.

Last editor of the synchronization set has insufficient rights

Meaning: The user specified (the last person to edit the synchronization set) does not have manager access to the database specified.

Solution: Either give the specified person manager access to the database or have an authorized manager edit the synchronization set document, thus becoming its 'owner'.

Multilingual databases can only be synchronized internally (not with other databases)

Meaning: Multilingual databases cannot be synchronized with unilingual ones.

Solution: Make sure your synchronization set contains one multilingual database or two or more unilingual ones.

No valid databases were successfully opened for this synchronization set

Meaning: None of the databases defined in the synchronization set could be opened.

Solution: Check that the databases have been built with the latest version of Domino Global WorkBench and that they contain synchronization features. Rebuild if necessary.

Non-original document encountered unexpectedly

Meaning: Internal database structure error.

Solution: Check that the database has been built with the latest version of Domino Global WorkBench. Rebuild if necessary.

Problem with Configuration db:

Meaning: A generic problem with the synchronization configuration database.

Solution: This is a general error message: more detailed message(s) will pinpoint the problem.

Too many document language versions were found in: <dbname>

Meaning: A document was found that exists in more languages than are supposed to be present in the database.

Solution: Check to see whether an API application, unaware of the Language Synchronizer, is creating documents.

Translation workflow error messages

This is an alphabetical listing of translation workflow error messages.

Error	Meaning
A Language Information document does not exist for the following language: %1	Translation workflow is implemented for databases that comprise a synchronization set. Every language in the synchronization set requires a language information document in the translation workflow database so that workflow can operate. These documents are normally created automatically when you save a workflow information document.
A name listed as Translation manager does not have the [TranslationMngr] role enabled in the database access control settings.	Users assigned as translation managers in the Workflow Assignments dialog must have the [TranslationMngr] role. Edit the database's access control list (ACL) to assign roles.
A required setup document could not be located: Workflow Information.	The workflow information document is fundamental to the workflow process. If it is missing you will have to re-create it and its language information documents.
A Workflow Information document already exists. Only one Workflow Information document can be created per database.	The translation workflow database should contain only one workflow information document.
A Workflow Information document has not yet been created. Would you like to create one now?	You will see this message if you open a translation workflow database that does not contain a workflow information document, which is fundamental to the workflow process.
An error occurred attempting to retrieve information stored in the Workflow Information document. This document may be corrupt.	The workflow information document is fundamental to the workflow process. If it is missing you will have to re-create it and its language information documents.
Assign Work: You have selected a document that you are not authorized to modify. You must have [WebMaster] role access in the ACL or be Translation Manager of all selected documents to perform this action.	In order to assign translation tasks for a language (in the relevant language information document) you must have the [Webmaster] role or be the translation manager for the language with role [TranslationMngr].
Cannot locate or open a database in the Synchronization Set. The server may be unavailable or the user may not have the proper ACL rights to the database: %1	Workflow setup error. Check that databases in the synchronization set are accessible.
Cannot locate required view in database: %1	Translation workflow relies on various views. Check that the specified view is present in the relevant database.
Cannot locate synchronizer configuration database: %1	The translation workflow database must be on the same server as the synchronization configuration database. This is also the server where the Language Synchronizer runs.
Corresponding document in Synchronizer database cannot be found.	A translation document (a document created by the Language Synchronizer), which has workflow in progress, is missing. The workflow agent mails the contents of the translator's working copy to the translation manager and marks the workflow tracking document Inactive. Provided the source document remains, the Language Synchronizer will re-create the translation document when it next runs, and workflow will restart.
Database does not exist, Synchronization may not be enabled for the Set Name defined in the Workflow Information document. Database: <name>	A database specified in the synchronization set document cannot be located. Check that you have selected the appropriate synchronization set document. Make sure the synchronization set document correctly identifies the databases to be synchronized.

Error	Meaning
Enter a comment to be appended to document history:	You will see this message if you open a workflow tracking document and click the 'Mark . . .' button.
Invalid Source Language entered in dialog	In most places you select language from a list, so you should not see this error. Enter a valid source language and retry. The source language is the language you are translating from.
Language Information documents have been created as responses. Work assignments need to be set up in these documents which can be located using the Setup view.	Data entered in Workflow Information form has been validated and saved. Next step is to set up work assignments.
Pasting of documents is not permitted in this database	You cannot paste documents into the workflow database.
Please have the Translation Manager check set up information. Missing information for the following database: %1	A required hidden document, (LevelsOfReview), could not be found. It is possible that the database is corrupted. You may have to setup workflow again.
Please have the Translation Manager check the language assignments. A workflow step has no name assigned to it.	One of the workflow steps has no-one assigned to it. Assignments for translation into a language are made in the language information documents for that language. You need the role [TranslationMngr] or [Webmaster] to edit this document.
Press OK to move all selected documents to the next workflow phase	You will see this message if you select documents in the My Work view of the translation workflow database and then click Phase Completed.
Press OK to move this document to the next workflow phase	You will see this message if you open a workflow tracking document and click the 'Mark . . .' button.
Press Yes to cancel the current workflow and restart translation using the modified source document. Press No to ignore the modified source and continue translating the existing document	If a source document changes while translation workflow is in progress, the workflow agent notifies the translation manager and gives him or her the option of canceling (restarting) workflow, or allowing the current translation to continue.
Profile document "Lng" not found in language database. Synchronizer may not be set up correctly: %1	Synchronization requires a profile document, \$Lng, to be present in each synchronized database. If the profile document is missing synchronization cannot operate.
The contents of an approved translation could not be processed because the destination language document could not be found	Translation is complete and approved, but the translation document (the document created by the Language Synchronizer) is missing. Save the translation in the translator's working copy by copying and pasting to another document and restart workflow (in the relevant workflow tracking document, click Restart Workflow).
The source for a document currently being translated has been deleted. The workflow will be canceled and working translations will be deleted. Document reference text: %1	A source document has been deleted while translation workflow is in progress. The workflow will be canceled.
The source for a document currently being translated has been modified. The Translation Manager must decide whether to continue or abandon the current translation. Document reference text: %1	If a source document changes while translation workflow is in progress, the workflow agent notifies the translation manager and gives him or her the option of canceling (restarting) workflow, or allowing the current translation to continue.
The Synchronization Set document for this database does not contain required language information.	Workflow setup error. The synchronization database may be corrupt. Try opening it in Notes. If that succeeds try workflow setup again.
The Synchronization Set you specified contains information about only one language. Translation Workflow requires at least two languages.	Workflow setup error. Check that the synchronization set (in the synchronization configuration database) includes the necessary languages.

Error	Meaning
The Synchronization Set you specified has no database documents	Workflow setup error. The synchronization set you selected has no database information documents. Language synchronization can only operate in databases that have corresponding database information documents in the synchronization configuration database.
This Translation Workflow application has been moved since it's Workflow Information document was created. Agent processing requires the database to be on the same server as the Synchronization Configuration database (synconf.nsf) that defines it's Synchronization Set. Agent processing cannot occur in replicas. Any changes should be replicated back to the original database for agent processing.	The translation workflow database must be on the same server as the synchronization configuration database. This is also the server where the Language Synchronizer runs.
Unable to locate a required synchronizer view in the language database	A required view could not be found. It is possible that the database is corrupted. You may have to setup synchronization again.
Unable to locate a required view in the workflow database	A required view could not be found. It is possible that the database is corrupted. You may have to setup workflow again.
What is the level of review required for this Workflow?	You must enter a value for 'Levels of review' on the workflow information document.
Working copy not found. It may have been deleted in language database.	The translator's working copy of a document cannot be located. restart workflow (in the relevant workflow tracking document, click Restart Workflow).
You do not have the correct roles in the database ACL to create or edit this document.	In order to edit the workflow information or language information document you must have the role [TranslationMngr] or [Webmaster]. Edit the database's access control list (ACL) to assign roles.
You must specify a valid Synchronization Set Name	Workflow setup error. You must select a valid synchronization set document.
You must specify the following number of Workflow States:%1 or else change the value of level of review. Default states have been entered.	On the workflow information document you should enter names for all your workflow steps. If you do not, default text will be used.

Index

Symbol

- .BMP files, 98
- @Commands
 - underscore characters in, 37
- @FormLanguage, 48
- @Functions, 48
- .GIF files, 98
- .JPEG files, 98
- \$Language field, 146, 150, 152, 154
- @LanguagePreference, 48
- \$Lng_CopyUNID field, 154
- \$Lng_LastModified field, 154
- \$Lng_LastSynchronised field, 154
- \$Lng_Original field, 154
- \$Lng_OriginalLang field, 154
- \$Lng_State field, 150, 152, 154, 179, 181, 183, 185
- \$Lng_SynchType field, 150, 154, 179
- \$Lng_UNID field, 154
- \$Lng_WorkflowState field, 154
- \$NavImageMap, 137

A

- Abandoning translation workflow, 177
- About documents, 139
- Access control list, 116
- ACL, 116
- Acronyms, 33
- Action buttons, 33
- Actions
 - tagging, 129
- Active languages, 67, 69, 87
- Adding new languages to a glossary, 68
- Address formats, 33
- Agent
 - translation workflow, 169
- Agents
 - aliasing, 37
 - in glossaries, 72
 - tagging, 130
- Aliases, 37, 125, 127
 - prefixes, 37, 106
 - underscore characters in, 37
- Aliases, creating or changing, 56
- Allow translation field, 61, 76
- Applets, 139
- Arabic, 16
- Assigning terms to term types, 71
- Attributes, 61, 76, 128

B

- Batch processing, 118

- Bi-directional languages, 16
- Bitmaps, 76
 - translating, 94, 98
- BMP files, 98
- Browsers, 18
- Building, 7
 - batch, 118
 - language databases, 27, 113, 114, 115, 116, 156
 - Notes Release 4.6 databases, 15
 - pseudo-language databases, 26
 - rebuilding databases, 123
 - selecting design elements for, 115
 - synchronized databases, 28, 29, 152, 156
- Built databases
 - testing, 26
- Bulleted text, 128

C

- Cancelling translation workflow, 177
- Cascaded views
 - display in tagged database, 141
- Changing the reference language of a glossary, 70
- Checkbox fields
 - keywords in, 39
- Collation
 - Scandinavian (in glossaries), 104
- Combine broken terms, 128
- Combobox fields
 - keywords in, 39
- Commenting glossaries for translation, 87, 88, 89, 92
- Comments
 - tagging, 130
- Concatenated sentences, 39
- Context fields, 61, 74
- Creating
 - design synopsis, 53
 - glossaries, 23, 64, 104
 - language databases, 113, 116
 - new terms in glossaries, 73
 - projects, 24, 102
 - pseudo-translations, 25, 82
 - synchronized databases, 28, 152, 156
 - tagged databases, 24, 106
 - term types, 70
- Currency formats, 33

D

- Data documents, 49, 134
- Database information document, 161

- Database titles
 - tagging, 130
- Databases
 - building, 27, 156
 - glossary, 7, 59
 - language, 6, 7, 167
 - modifying after localization has started, 52
 - multilingual, 6, 7, 47, 48, 150
 - preparing, 36, 37, 39, 40, 41, 42, 43, 44, 46, 49, 50, 51, 52, 53, 55, 56, 168
 - project, 19, 102
 - rebuilding, 123
 - source, 4, 7
 - synchronized, 7, 48, 150, 152, 154, 156, 163
 - Synchronizer configuration, 161
 - tagged, 6, 7
 - unilingual, 6, 7, 150
 - workflow, 167
- Defining language databases, 114
- Deleting
 - duplicates in a glossary, 81
 - non-linguistic conflicts in a glossary, 81
 - orphan translations in a glossary, 81
 - tags, 143
 - terms in a glossary, 80
- Design considerations, 33, 36, 37, 39, 40, 41, 42, 43, 44, 46, 49, 50, 51, 52, 56, 150
- Design synopsis, 55
 - creating, 53
- DGWPROJ.NTF, 19, 102
- DGWSTORE.NSF, 68, 152
- Dialog boxes
 - Select Work Assignments, 169
- Dialog list fields
 - keywords in, 39
- Discussion, 21
- DNT (do not translate) text, 87, 88, 89, 92, 93, 98
- Documentation, 21
- Documents
 - database information
 - (synchronization), 161
 - global, 150
 - help (data), 49, 134
 - language information, 169
 - local, 150
 - profile, 12
 - source (translation workflow), 167, 177
 - synchronization set, 161

- translatable, 150
- translation (in glossaries), 62
- translation (translation workflow), 167, 177
- workflow information, 169
- workflow tracking, 169, 174, 177, 179, 181, 183

Domino, 18

Domino Global WorkBench

- upgrading from earlier releases, 13

Dynamic variables, 41

E

Error messages, 187, 205, 217

Euro symbol, 33

Excluding text from translation, 44, 113

Exclusion list, 106, 113

Expansion of text, 33

Exporting, 77, 78

F

Fields

- handling field names, 39
- in synchronized databases, 154, 179, 181, 183, 185
- keywords in, 39
- tagging, 130

Flag icon, 116

Fld_touch field, 106

Folders

- aliasing, 37
- tagging, 132

Font mapping, 116

Fonts, 33

- changing on building, 12, 116

Forcing an update, 123

Formats

- international, 33

Forms

- aliasing, 37
- tagging, 132

Formulas, 40

- tagging, 125, 127, 132

Framesets

- tagging, 133

Functions, 48

G

GIF files, 98

Global documents, 150

Global glossaries, 59

Globalization, 21

Glossaries, 7, 59, 61, 63, 67, 68, 69, 70

- agents in, 72
- changing priority of, 104
- commenting for translation, 87, 88, 89, 92
- creating, 64, 104
- creating new terms in, 73

- deleting terms in, 80, 81
- duplicate entries in, 100
- error messages, 205
- exporting terms from, 77, 78
- global, 59
- importing terms to, 46, 77
- introduction, 4
- loading, 104
- local, 59
- managing, 72
- non-linguistic conflicts in, 81
- obsolete terms in, 81
- of Lotus terms, 59
- order in which WorkBench consults, 104
- orphan translation documents in, 81
- planning, 63
- preparing, 23
- priority, 104
- project, 59
- replicating, 84
- replication conflicts in, 81
- retrieving terms into, 79
- reviewing, 25, 71
- roles in, 72
- Scandinavian collation in, 104
- translating, 71, 87, 88, 92, 93
- translation documents, 62
- upgrading from NGD 4.6, 46, 72

Glossary marking, 106

Graphics, 33, 76, 96

- translating, 94, 98

Guidelines for Developers and Translators, 21

H

Hebrew, 16

Help (data) documents, 49, 134

Help About documents, 139

Help Using documents, 139

Hidden elements, 43

- tagging, 134

Hide when formulas, 41, 128

HTML

- and bi-directional languages, 16
- in source databases, 44
- tagging, 135

I

Icons

- project, 103

Identifiers

- language, 68

Images

- tagging, 139

Importing, 46, 77, 79

Information documents

- for translation workflow, 169

Installation, 19

Installing the Language Synchronizer, 156

International considerations, 33

Introduction, 1, 3, 4, 7, 12, 15, 18, 19

J

Java resources, 45, 139

- localizing, 12

JavaScript, 42

- tagging, 136

JPEG files, 98

K

Keywords, 39

- aliasing, 37
- tagging, 130

L

Language databases, 6, 7, 167

- building, 3, 27, 113, 114, 115, 116
- defining, 114
- design changes in, 144
- functionality in, 143
- layout in, 116
- Notes Release 4.6, 15
- testing, 26, 82
- updating, 27, 119, 120, 122

Language identifiers, 68

Language information document, 167

- for translation workflow, 169

Language SwitchBar, 150, 152, 156

Language synchronization, 12, 149, 152, 154, 156

- configuration database, 150, 161
- sets, 156

Language Synchronizer, 150

- installing, 156

Languages

- adding to a glossary, 68
- making active, 67, 87
- making inactive, 69
- switching in built databases, 146

Layout

- preserving in language databases, 116, 120

Length limitations, 50

Listbox fields

- keywords in, 39

Loading

- glossaries, 104
- source databases, 103

Local documents, 150

Local glossaries, 59

Localization, 1

- analysis, 10
- tasks, 10

Location fields, 61, 74

LotusScript, 41

- disabling recompilation of, 116

in Notes Release 4.6 databases, 15
recompilation of, 115, 116
tagging, 125, 127, 136

M

Main documents, 61
Making a backup, 53
Making languages active, 67
Making languages inactive, 69
Marker text (in glossaries), 106
Marking terms as obsolete, 81
Menus
 pop-up, 101
Messages, 187, 205, 217
Multilingual databases, 6, 7, 47, 48, 146, 150

N

Name formats, 33
Native images, 98
 tagging, 139
Navigators
 aliasing, 37
 tagging, 137
 translating graphics from, 98
Non-linguistic conflicts
 in glossaries, 81
Notes Global Designer
 upgrading from, 14
Notes Release 4.6 databases, 15, 16

O

Obsolete terms, 61, 81
ODS, 144
On-disk structure, 144
Opening projects, 103
Orphan translation documents
 in glossaries, 81
Overview, 1, 3, 4, 7, 12, 15

P

Pages
 tagging, 138
Plain text, 61, 76, 128
Planning glossaries, 63
Pop-up menus, 101
Preferred language, 146
Prefixes
 for aliases, 37, 106
Preparing glossaries, 23
Preparing source databases, 23, 33, 36, 37, 39, 40, 41, 42, 43, 44, 46, 47, 49, 50, 51, 52, 53, 55, 56, 150, 152, 168
Preserving layout
 in language databases, 12, 116, 120
Preventing translation, 44, 87, 88, 89, 92, 93
Priority
 of glossaries, 104

Private design elements
 tagging, 139
Process
 for creating synchronized databases, 28, 29
 for localization, 23, 24, 25, 26, 27
 for translation workflow, 30, 31
 overview, 7
Processes
 recommended, 23, 28, 30
Profile documents, 12
PROJECT.NSF, 19, 102, 116
Project database, 19, 102
Project glossaries, 59
Project icons, 103
Project Manager, 12
Projects, 3, 7, 23, 24, 25, 26, 27, 101, 102
 creating, 24, 102, 103
 opening, 103
Prompting
 when creating tagged database, 106
Pseudo-language databases
 building, 26
 testing, 26
Pseudo-translations
 creating, 25, 82

R

Radio button fields
 keywords in, 39
Re-installing, 19
Rebuilding databases, 123
Recommended processes, 23, 28, 30
Recompilation of LotusScript, 115, 116
Reference language, 63
 changing, 70
Removing tags, 143
Replace fonts, 116
Replicating glossaries, 84
Replication
 conflicts in glossaries, 81
 of synchronized databases, 185
 of translation workflow databases, 185
Resources
 tagging, 139
Retrieving translations, 79
Reviewing glossaries, 25
Rich text, 61, 76, 96, 128
Right-to-left reading (RTL) text, 16
Roles, 10
 in glossary databases, 64, 72
 in translation workflow, 168
 tagging, 139
RTL text, 16

S

Scandinavian collation
 in glossaries, 104

Scheduling the translation workflow
 agent, 169
Script libraries, 41, 139
Select Work Assignments dialog, 169
Selecting items
 for building, 115
 for tagging, 106
Shared / Personal on first use folders or views
 tagging, 140
Shared / personal on first used folders or views, 44
Shared design elements, 46
Shared fields
 tagging, 139
Simple actions, 51
Source databases, 4, 7
 loading, 103
 modifying after localization has started, 52
Notes Release 4.6, 15, 16
 preparing, 23, 33, 36, 37, 39, 40, 41, 42, 43, 44, 46, 47, 49, 50, 51, 52, 53, 55, 56, 168
Source documents
 translation workflow, 167, 177, 179, 181, 183, 185
Special folder and form names
 tagging, 140
Static text
 tagging, 140
Status (glossary), 62
 approved, 72, 96
 prevent translation, 98
 translated, 72, 96
 untranslated, 96
Stopping translation workflow, 177
Structured text files, 77, 78
Subforms, 52
 tagging, 140
SwitchBar, 150, 152, 156
Switching languages, 146
Synchronization
 see Language synchronization, 12, 150
Synchronization set document, 161
Synchronized databases, 7, 28, 48, 150
 building, 28, 29, 152, 156
 fields in, 154
 preparing, 152
 replicating, 185
 updating, 120, 156
 upgrading, 13
 using, 163
 views and forms in, 152
Synchronizer
 see Language Synchronizer, 150, 156
Synchronizer configuration database, 161
 installing, 156
System requirements, 18

T

- Tag delimiters, 106
- Tagged databases, 6, 7
 - creating, 3, 24, 106
 - updating, 27, 119, 120, 122
- Tagger, 12
- Tagging, 124, 129, 142
 - actions, 129
 - agents, 130
 - comments, 130
 - database titles, 130
 - excluding text from, 106, 113
 - fields, 130
 - folders, 132
 - forms, 132
 - formulas, 125, 127, 132
 - framesets, 133
 - hidden design elements, 134
 - HTML, 135
 - images, 139
 - in general, 125
 - Java projects, 3
 - JavaScript, 136
 - limitations of, 141
 - LotusScript, 125, 127, 136
 - native images, 139
 - navigators, 137
 - options, 24
 - pages, 138
 - private design elements, 139
 - references to other design elements, 50, 127
 - resources, 139
 - reviewing results of, 113
 - roles, 139
 - shared / personal on first use folders or views, 140
 - shared fields, 139
 - special folder and form names, 140
 - static text, 140
 - subforms, 140
 - tags, 125
 - URLs, 141
 - views, 141
 - when using glossaries upgraded from 4.6, 14
- Tagging Options dialog box, 106
- Tags, 125
 - removing, 143
- Template Globalization Best Practices database, 21
- Term definition document, 61, 73
- Term document, 61, 73
- Term types, 61, 70
 - assigning terms to, 71
 - creating, 70
 - using, 71
- Testing built databases, 26, 82
- Text expansion, 33
- Tips, 21

- Tracking documents
 - in translation workflow, 167, 174, 179, 181, 183
- Translatable documents, 150
- Translated document
 - translation workflow, 167
- Translating, 27, 76, 87, 88, 89, 92, 93, 94, 96, 98, 100
 - graphics, 98
- Translation, 1, 7, 71
 - preventing, 44, 87, 88, 89, 92, 93
- Translation documents
 - in glossaries, 62
 - in translation workflow, 167, 177, 179, 181, 183, 185
- Translation status, 62
- Translation workflow, 7, 12, 30, 31, 149, 166, 168, 174, 178, 179, 181, 183, 185
 - cancelling, 177
 - database, 167, 169
 - deleting documents, 177
 - error messages, 217
 - information document, 167, 169
 - language information document, 169
 - replicating databases, 185
 - tracking document, 167, 174, 179, 181, 183
- Translation workflow agent
 - scheduling, 169
- Translations
 - exporting, 78
 - importing, 77
 - retrieving from another glossary, 79
- Troubleshooting, 21
 - failure to find glossary text, 104
 - in built language databases, 143
- TSA (translate same as) text, 88

U

- Underscore characters in aliases, 37
- Unicode, 19, 33
 - display in WorkBench, 12, 19, 101
- Unilingual databases, 6, 7, 146, 150
- Updating, 3, 119, 120
 - batch, 118
 - forcing an update, 123
 - language databases, 27, 119, 120, 122
 - source databases, 52
 - synchronized databases, 120, 156
 - tagged databases, 27, 119, 120, 122
 - WorkBench display, 52, 120
- Upgrading
 - glossaries, 46, 72
 - Notes Global Designer 4.6 projects, 14
 - projects from earlier releases, 13
 - synchronization, 13
- URLs
 - tagging, 141
- Using
 - WorkBench, 101

- Using documents, 139
- Using synchronized databases, 163
- Using term types, 71

V

- Views
 - aliasing, 37
 - cascaded, 141
 - for translators, 94, 100
 - tagging, 141

W

- Warning messages, 187, 205, 217
- Web
 - browsers, 18
 - navigators, 137
 - sites, 6
- What's new?, 12
- Who does what?, 10
- WorkBench
 - error messages, 187
 - introduction, 3
 - restarting after a crash, 187
 - starting, 24
 - using, 101
- Workflow
 - translation, 12, 30, 149, 166, 167, 168, 169, 174, 178, 179, 181, 183, 185
- Workflow agent
 - scheduling, 169
- Workflow information documents, 169
- Workflow tracking documents, 167, 169, 174, 177, 179, 181, 183