# Using the AuthorIT Localization Manager

By AuthorIT Software Corporation Ltd.

**AuthorIT Software Corporation**
PO Box 300-273
Albany
Auckland
New Zealand
Telephone +64-9-915 5070
Fax +64-9-915 5071
feedback@author-it.com
http://www.author-it.com

# Contents

## Chapter 5  Managing Updates to Content <span style="float:right">47</span>

## Glossary of Terms <span style="float:right">55</span>

## Index <span style="float:right">57</span>

Chapter 1

# The AuthorIT Localization Manager

Many companies produce documentation in multiple languages, and as part of this they are faced with having to invest considerable money and energy in translation and localization efforts. AuthorIT's *Localization Manager* makes this process more efficient reducing the cost and time involved.
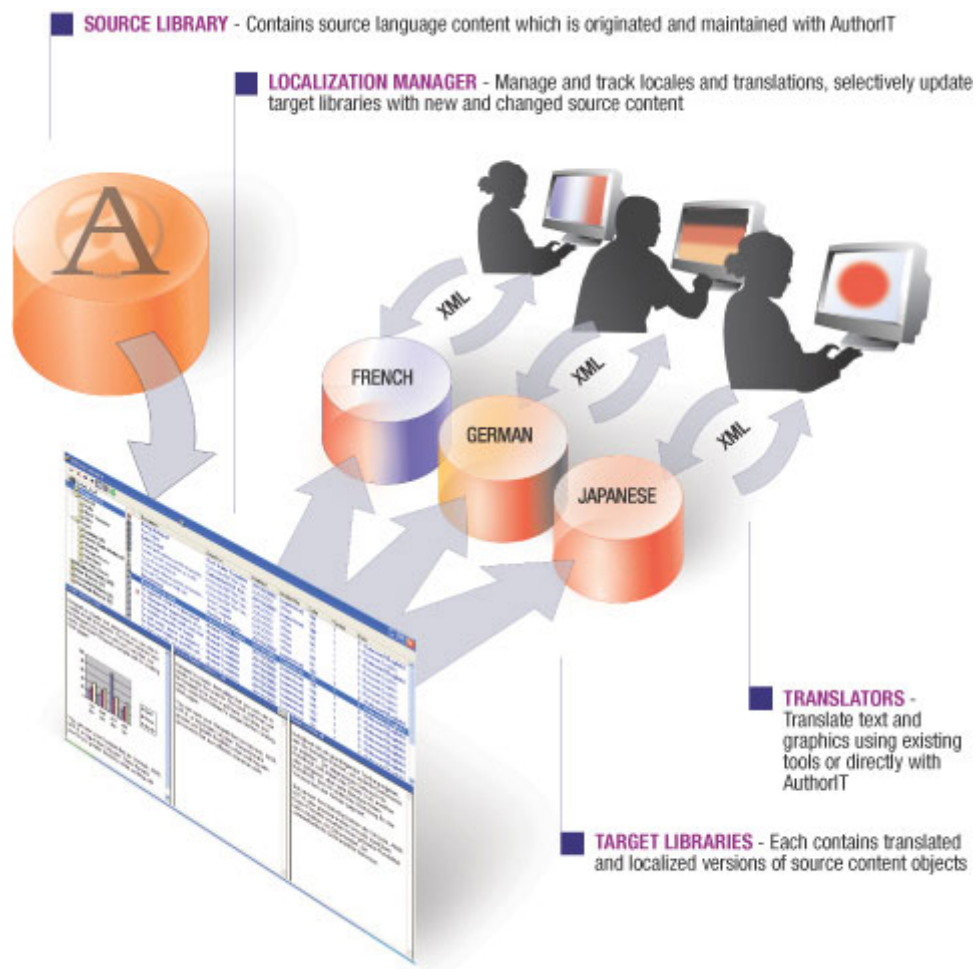
**Enterprise Edition**

Localization Manager provides you with the functionality to manage the creation, comparison, and synchronization of your content in multiple languages. It has the ability to export and import any content as XML also allowing for integration with translation memory (TM) tools like TRADOS, Deja Vu, Star Transit, SDLX, to name a few.

## In This Chapter

# Localization, Globalization, Internationalization… What Does it Mean?

*Localization, globalization, internationalization, translation...* All terms and processes that easily create *confusion!*  So what does it all mean?

Let's take the example of a popular North American fast-food chain. It expands and opens new chains worldwide, across Europe, South America, Asia, Australasia... This process is known as **globalization**.

By thinking ahead, the company plans and creates its products so they can easily be adapted to suit different locales. As part of the expansion, the fast food chain may adapt the menu to cater for differing local tastes or cultural needs.  Replacing beef products with chicken or lamb products in Indian countries, or one soft-drink brand with another in others. This process is known as **internationalization**.

**Localization** refers to adapting a product or service to a particular language, culture, or other needs of the target market or 'locale'. In the majority of cases, a large part (and cost) of localization involves the **translation** of content from one language to another.

However, localization is more than just translating text. Other considerations for localization include:

▸ Ensuring the appropriate regional standards such as measurement terms (metric vs. imperial), Date/Calendar formats (mm/dd/yy vs. dd/mm/yy), paper size (A4 vs. Letter) and so on, are used.

▸ Including the correct legal requirements in a warranty statement, or the appropriate contact information and telephone number of a technical support department.

▸ Possibly replacing diagrams or graphics, particularly those that contain text or a screenshot of a differing user interface

▸ Recognizing any factors that may be specific to a culture or country. Take for example a traffic light (also referred to as a 'stop light'). In the US these lights are red, orange and green. However in Japan, while the colors look the same, they translate to being red, orange and blue...

As you can see, the list goes on. Use the wrong term or the wrong graphic, and your viewers can be left scratching their heads in confusion, or highly offended by your seemingly innocent mistake.

# The Cost of Localization

Localization can be a complicated and expensive project. One mention of localization and the immediate reaction from your financial department may be to reach defensively for their wallet. Costs can be unpredictable and can quickly get out of control, particularly if you don't know what to expect. You may choose to manage the translation in-house, or to outsource it to an external company that specializes in localization and translation.

In simple terms, translation is not a cheap task. Let's look at an example to put this in perspective:

The average cost a translator will charge is around 20 cents (US) per word. Take a document with 1000 topics and an average of 200 words per topic. That's 200,000 words so you're quickly looking at *$40,000*. Now remember that's just for the initial translation. The growing cost comes when you make modifications to the original document and need it retranslated. Most translation agencies use *translation memory tools* (see "How Do Translation Memory Tools Work?" on page 9) which help reduce the effort involved in retranslating a document, but they still charge for the whole document (albeit at a reduced word rate for the text already translated).

When using translation memory tools, a **fuzzy match** is returned where a text string is similar but not identical. An **exact match** (100%) is returned where there is no difference or variation between the two strings.
Translators often charge different rates when text is found as an exact match, as a fuzzy match (with the match falling between a certain percentage), or is a new translation.

Let's get back to our example. You now modify only 5 of these topics. Without allowing for *fuzzy matches*, the cost of retranslation can quickly climb to $16,120.00 :

| | |
|---|---|
| 5 topics with 200 words @ 20 cents per word | = $200.00 |
| 995 topics with 200 words @ 8 cents | = $15,920.00 |
| Total cost of retranslation | = $16,120.00 |

And remember that's just to translate *5* modified topics!

Over time, these costs quickly mount up. Our example was just one document into one language. Translate that same document into 10 other languages, and multiply the cost 10 times. Translate a further 10 or 100 documents into multiple languages, and watch your costs skyrocket!

### How AuthorIT Reduces Translation Costs:

Because of the unique way AuthorIT stores and manages content, savings are quickly realized:

**1)**   You only translate objects that have been modified.

For example, let's go back to our 1000 topic document which we've now updated. Rather than sending the translator all 1000 topics again, only the 5 modified topics are exported as XML. Using our previous example this would reduce the cost of retranslation from $16120.00 to $200.00!

**2)**   Text is only translated once.

One of AuthorITs biggest strengths is reusability. The same topics are reused in multiple documents. For example, the same Copyright Notice (or even an entire Introductory chapter) may be used in many Books. Each topic only requires translation once. Embedded topics take reuse even further, and again are only translated once.

Cross references and related topic groups don't even require translation. Because these are inserted at publishing time, taking their text from the heading of the topic they reference, they aren't stored in the topic text making less text to translate. Likewise, reference text such as `See` and `on  Page` are defined by the Hypertext link object template used, so only the template requires translation.

Our studies have shown an average of a *30% reduction in word count* because of reuse.

**3)**   The XML files do not contain formatting.

When the same text string is found using different character formatting, memory translation tools do not always identify it as an exact match. As the XML files do not contain formatting, this helps to increase the exact matches found.

## Should I Manage the Localization Project In-house or Outsource It?

Whether you:

**a**   manage the entire project and translation in-house,

**b**   manage the project in-house but outsource the translation, or

**c**   share the project with a localization vendor and have them manage the translation,

will all depend on the resources, budget, time constraints, and expertise you have available.

Consider first the role of the Localization Coordinator. This is the person (or group) in charge of coordinating the translation, and managing the updates to translated content.

Whichever option you choose, it is important that the Localization Coordinator gives priority to the localization. While they may have other duties, particularly if the project is managed in-house, these duties should come second to their localization responsibilities.

# Roles and Responsibilities in the Localization Process

As a general rule, most companies have the following groups contributing to the creation of content and the localization process:

| Role | Responsible For |
|---|---|
| Authors | Creating, modifying, and maintaining the content in the main AuthorIT library. |
| Reviewers | Reviewing AuthorIT content. |
| Localization Coordinator | Coordinating and managing the localization process, including overseeing the translation. |
| | This role may be shared with a translation company or companies. |
| Translators | Translating content into the target language. |
| Localization Reviewers | Reviewing translated content. |

While the workflow and specific responsibilities may differ from one company to the next, the roles are generally very similar.

# How Do Translation Memory Tools Work?

It's important to understand that translation memory tools are quite different and far more sophisticated than machine translation tools.

## Machine Translation

Many of us have used Internet translation similar to that provided by the Language Tools on Google or AltaVistas Babel Fish in an attempt to translate an email or unknown text. We quickly realize that while it may give us an idea of the underlying message and prove to be a quick low-cost solution, it is certainly not perfect.

While this technology is useful if you get a document or email in German and want to know what is says, the results are far from publication quality. It is also limited in the languages it supports.

Some time ago, we received an unsolicited email offering to translate our entire website into 7 languages automatically. It even had an example of our front page. On this page we have a strap line, "Affordable Content Management". One of our staff speaks Chinese and translated the example they provided - he looked at it for a while then said "Cheap happy management". Not exactly the message we were looking for. Needless to say we did not take up their offer.

So as you can see, machine translation is no match for a real person!

## Translation Memory Tools

A segment is a string of text, usually a sentence, but in some languages a segment may be a whole paragraph.

Translation memory tools store previously translated **segments** in a database, enabling translators to work more efficiently and reducing the need for retranslation. When translating a new document, these tools find and retrieve identical or similar matches of these segments for reuse. The results are returned with a rank or score according to the percentage of similarity between the text being translated and the match found.

▸ An **exact match** (100%) is returned where there is no difference or variation between the two strings.

▸ A **fuzzy match** is returned where the strings are very similar but not identical. The same string that uses different formatting can return a fuzzy match.

Translators often charge different rates when text is found as an exact match, as a fuzzy match (with the match falling between a certain percentage), or is a new translation.

Chapter 2

# Overview of the Localization Process in AuthorIT

Each localization is stored in a separate AuthorIT library known as a **Target Library**. Localization Manager then controls the flow of information (new and modified content) from the Source Library to the Target Libraries, and manages the translation of that content.

Target Libraries are essentially created as a copy of the Source Library. Content is then translated in the Target Library in one of two ways:

1) Directly in the Target Library with objects being opened, modified and saved using AuthorIT.

   -or-

2) Translating the content externally by:

   1. Exporting the translatable content as XML from the Target Library.

   2. Translating the XML with a translation memory tool (such as TRADOS).

   3. Importing the translated XML into the Target Library, overwriting the original content.
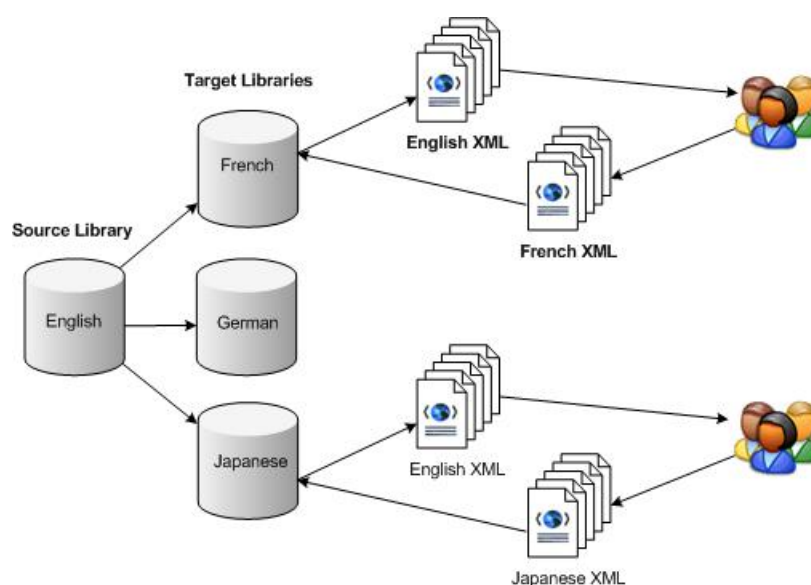


When changes are made in the **Source Library**, updates can be sent to the **Target Libraries** and the new and modified content examined and selectively imported using the *Library Compare* (see "Using the Compare Update Utility" on page 49) functionality.

The localization process within AuthorIT can be broken into three main areas:

1) **Setting it up:**

Defining the localizations you will work with and creating a Target Library for each localization ready for translation.

**2)   Translating the Content:**

Translation can be done directly in the Target Library, but in many cases it is outsourced to an external agency. Localization Manager is used to export content as XML for translation, then to import it when translation is complete.

**3)   Managing updates to the content:**

Translation is not just a one off process and any updates to the content in the original library normally require modification and retranslation in the target language. By reusing content, the amount of data requiring retranslation is reduced. Rather than sending the translators an entire document, only the content that is changed is sent.

## In This Chapter

# The Interface of the Localization Manager

This section describes the work environment and the interface elements you will encounter while you're using the Localization Manager

## The Main Window

The Main window displays two panes. The left pane contains the libraries, localizations, updates and jobs in an easy to understand hierarchy. When something in the left pane is highlighted, the right pane shows additional details. The buttons on the toolbar are enabled and disabled to reflect what may be done with the highlighted objects. These same functions are available by right clicking.
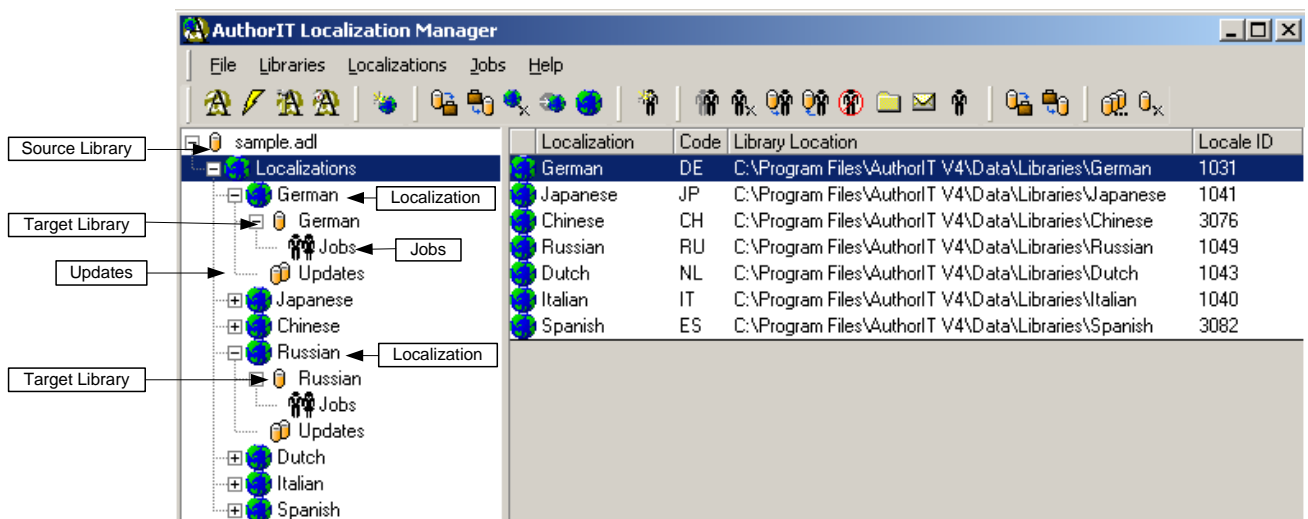
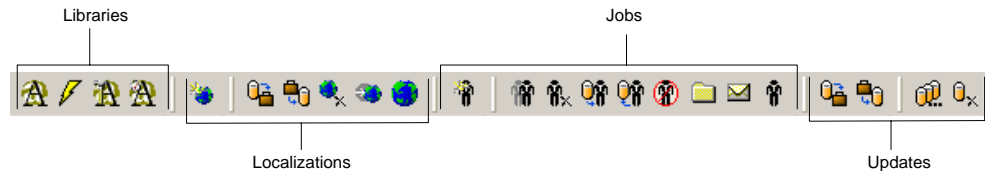Certain menu options and toolbar buttons are only available when in a particular folder. For example, you must have the Jobs folder selected before you can add or work with a Job.

# The Localization Manager Toolbars

The Main Toolbar is made up of four separate toolbars:



## Libraries Toolbar

Use the Libraries toolbar to quickly open the Source or Target Libraries with other AuthorIT modules.

| This Button | Does this |
| --- | --- |
| | Opens Library with the main AuthorIT application |
| | Opens Library with the AuthorIT Publisher |
| | Opens Library with the AuthorIT Administrator |
| | Opens Library with the AuthorIT Project Manager |

Functions available on this toolbar, are also available under the **Libraries** menu.

**Note:** These buttons are only enabled when a Library is selected.

## Localizations Toolbar

The Localizations Toolbar gives you quick access to commonly used commands when creating or working with localizations.

| This Button | Does this |
| --- | --- |
| | Creates a new localization |
| | Creates an updates for **all** localizations |
| | Creates a Library Update |
| | Unpacks a Library Update |
| | Deletes the localization |
| | Creates a copy of the selected localization |
| | Displays the Properties of the localization |

Functions available on this toolbar, are also available under the **Localizations** menu.

> **Note:** These buttons are only enabled when in the Localizations or Localization folder.

## Jobs Toolbar

Use the Jobs toolbar to create new Translation Jobs or work with existing Jobs.

| This Button | Does this |
| --- | --- |
| | Creates a new Job |
| | Copies the selected Job |
| | Deletes the selected Job |
| | Exports the selected Job to XML |
| | Imports the XML of the selected Job |
| | Imports the XML of the selected Job in draft format. The content is updated but the objects remain locked. |
| | Cancels the selected Job |
| | Opens the folder containing the XML files |
| | Starts an Email message attaching the zipped XML files |
| | Displays the properties of the selected Job |

Functions available on this toolbar, are also available under the **Jobs** menu.

> **Note:** These buttons only become enabled as the Job moves through its various stages and the option is available. For example, the Import Jobs button is disabled until the Job has been exported.

## Updates Toolbar

The Updates Toolbar gives you quick access to commonly used commands when creating or working with Updates.

| This Button | Does this |
| --- | --- |
| | Creates a Library Update |
| | Unpacks a Library Update |
| | Opens the Compare Update utility |
| | Deletes the selected Update |

> **Note:** These buttons are only enabled when the Updates folder is selected and that option is available. For example, the Compare Update button does not display if there is not an existing Update.

# Evaluation Limitations for the Localization Manager Module

An evaluation version of Localization Manager is automatically included with the AuthorIT Evaluation edition, and with the fully licensed Enterprise edition.  The Evaluation version allows you to:

‣  create up to 2 localizations

‣  create up to 5 translation jobs for each localization

‣  include up to 30 objects in each translation job

‣  create up to 10 updates per localization

This will allow you to complete the tutorial and evaluate the features. To make full use of Localization Manager features, you need a Localization Manager license. When you are ready to purchase a Localization Manager license, contact your local AuthorIT Reseller, or *order directly from us* http://www.author-it.com/index.mv?buy.

Chapter 3
# Setting it Up

Before you can begin translating content, you need to set up the localizations you want to work with, and create a Target Library for each localization. You should also set up release states to control the flow of information, and to identify where in the localization process each component is.

This section discusses this in detail.

## In This Chapter

# Defining Release States and User Groups for Localization

Before creating localizations, give consideration to the release states and user groups you will need. By creating the release states in your Source library, they are automatically included in each Target Library. You can then use the release states to manage your workflow and effectively create a handover of an object from one group to another.

We suggest the following release states as a guideline:

- Draft (Source)
- Review (Source)
- Released (Source)
- Sent for Translation
- Translation Review
- Released (Translated)

Of course, these are only examples and you can create your own customized release states to show where, in the localization cycle, a particular component is.  For more information on setting up release states, refer to the **Using AuthorIT Administrator Guide**.

For example, you may use the following workflow:

| Role | Responsibility |
|---|---|
| Authors | Create content in **Draft (Source)**, when complete set to **Review (Source)**. |
| Reviewers | Review objects in **Review (Source)**, if accepted, set to **Released (Source)**, if rejected return to **Draft (Source)** with comments in an annotation style. |
| Localization Coordinator | Receive objects as **Released (Source)**, export to XML and set state to **Sent for Translation** |
| Localization Coordinator | Receive translated XML, import and change objects to **Translation Review**. |
| Localization Reviewers | Review translated objects in **Translation Review** state and, if accepted, set state to **Released (Translated)**. If rejected (and this is the important part) revert the object to the English version and set the state back to **Released (Source)**. |

When creating release states there are two color attributes to define: an **object color** and a **review color**. When a review color is defined, the objects in that release state are highlighted with the review color in the published Word output as a visible indication that they need reviewing.

# Adding a Localization

Having defined the release states you'll need, you then need to create a **Localization** for each language variant you require.

Each Localization has the following attributes:

➤ **Localization Name:**  A description used to identify the localization you are working with, for example "French", "German", etc. This also ends up being the name of your Target library.

➤ **Locale ID:**  A unique identifier with a 32-bit value defined by Microsoft Windows. For example English (US) is 1033.

➤ **Library Path:**  Location of the directory where the Target library will be created. By default this path is set to a create a subfolder under your main Libraries folder (set by the Libraries path defined in AuthorIT under **Tools > Options > File Locations**).

**Note:** The default path sets the name of the subfolder based on what is first entered in the Localization Name field. If you later change the Localization Name, the path is *not* updated.

➤ **Abbreviation:**  A two character abbreviation used to identify the localization. By default this property is preset to the Windows Locale Code of the **Locale ID** chosen. For example, "EN" for English, "FR" for French, and so on.  This abbreviation is then displayed beside each object when viewing folders.

➤ **Object Inclusion Rules:**  A rule, or set of rules, covering which objects will be included in the Target library when it is created.
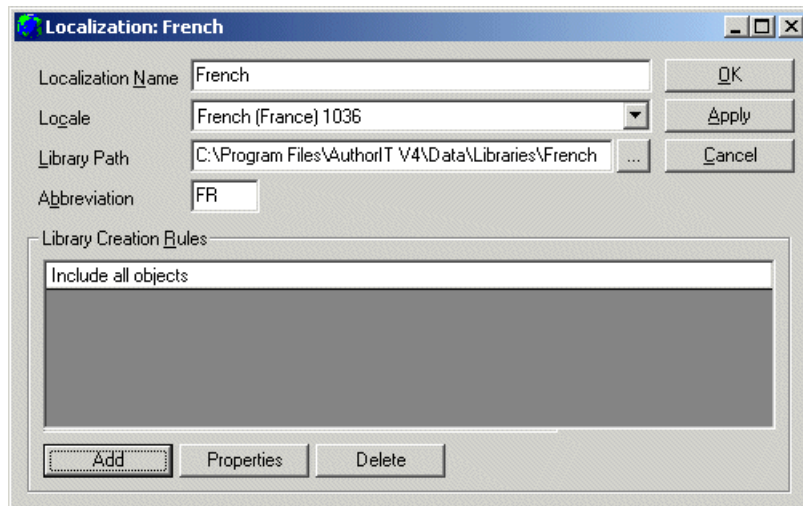


## *To Add a Localization:*

1) Open the Source Library with Localization Manager

2) Select the **Localizations** folder

3) Choose the **Add Localization** button on the main toolbar

   -or-

Right click and choose **Add Localization**
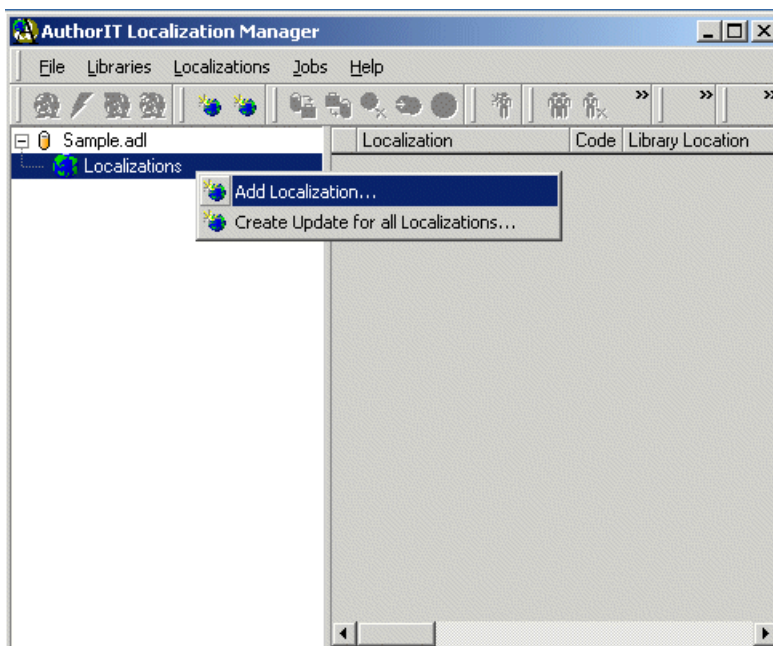


4)   Enter the locale attributes and *object inclusion rules* (on page 21) then choose **OK**.
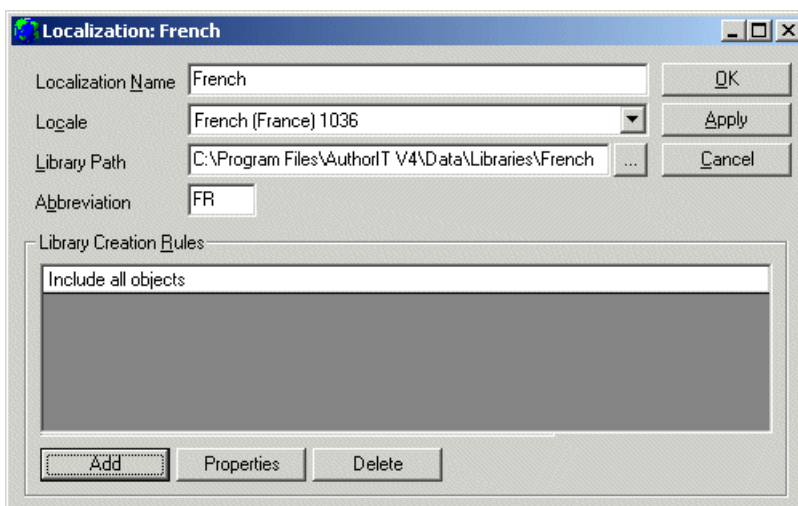
**Tip**: Once you have created a Localization, you can *copy it* (see "Copying a Localization" on page 22) to create further Localizations. Copying a Localization is faster than creating a new Localization from scratch, as the rules you have already defined are also copied.

The Localization is created and now appears in Localization Manager.
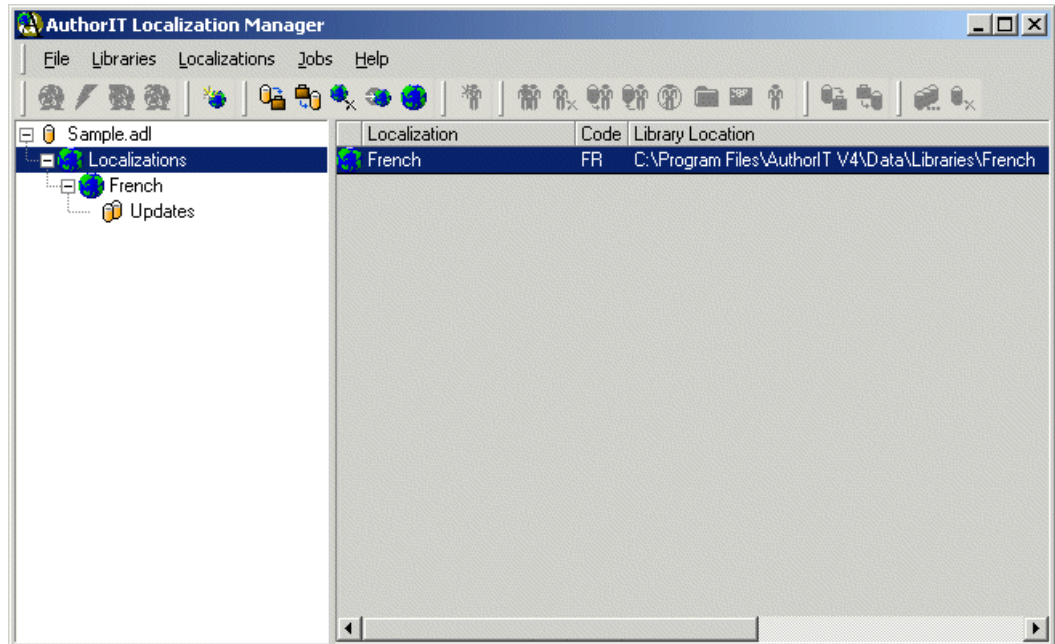


# Object Inclusion Rules

Object Inclusion Rules are used to define a set of criteria that control which objects are included or excluded in the Target Library.

There are many instances when a number of objects don't require Localization or when you want to restrict the information that is included in the Target Library. For example, your Library may contain sensitive information that should not be available outside the company and/or you may only need to translate the end-user documentation that will be distributed to your viewers.

### *To Add Object Inclusion Rules:*

**1)** Open the Localization properties and choose **Add**.

The **Object Inclusion Rule** window appears
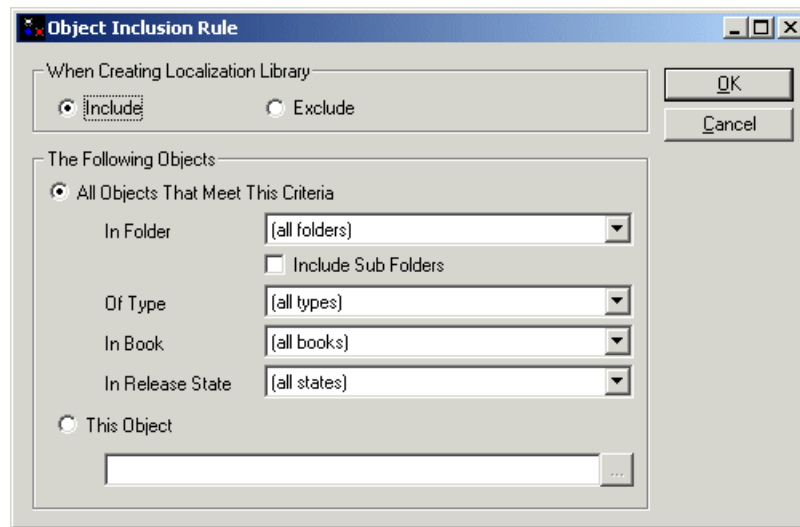


**2)** Enter your criteria to include or exclude objects and choose **OK**.

**3)** Repeat these steps to create additional rules.

# Copying a Localization

Copying an existing Localization is faster than creating a new one and setting its properties from scratch as the rules you have already defined are also copied. For example, you may have already created a Localization with a number of rules that include all objects used by the Books that make up your end user documentation, but that exclude specific objects that should not be available outside the company. By copying the Localization, these rules are also copied and you don't need to create them all over again.

### To Copy a Localization

**1)** Select the Localization that you want to copy

**2)** Choose the **Copy Localization** button on the main toolbar

-or-

Right click and choose **Copy Localization**



**3)** A message appears asking you to confirm the action
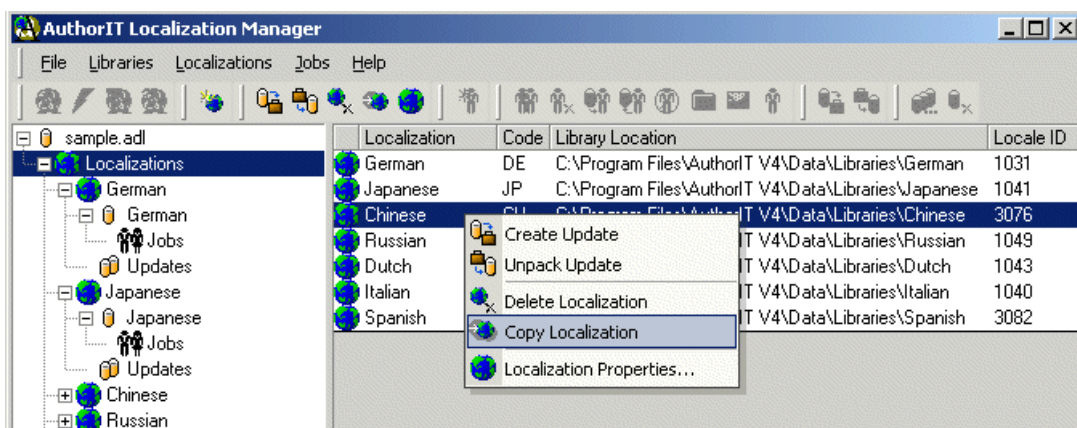
**4)**  Choose **Yes** to create a copy of the Localization. AuthorIT creates and displays a duplicate of the original Localization with "Copy of " inserted at the beginning of the description.

---

**Note:**  The default Library path sets the name of the subfolder based on what is first entered in the Localization Name field. If you don't rename the Localization first, the path is *not* updated.

---

**5)**  Change any settings, including the new Localization's Description, then save your changes.

# Creating a Target Library

Each Localization requires a **Target Library**.

A Target Library is a complete library containing objects that mirror those in the Source Library. Objects share the same object key but are completely "localized". Objects can then be translated, either directly in the Target Library, or externally and then imported.

Creating a Target Library is very similar to creating an offline library, except than any linked files are included with the Target Library as linked files rather than being embedded as an external resource. This allows for these files to also be localized where necessary.

## *To Create a Target Library:*

**1)**  Open the Source Library with Localization Manager

**2)**  Select the Localization you want to create a Target Library for

**3)**  Choose the [icon] button on the main toolbar

-or-

Choose **Localizations > Create Update** from the main menu

-or-

Right-click and choose **Create Update**
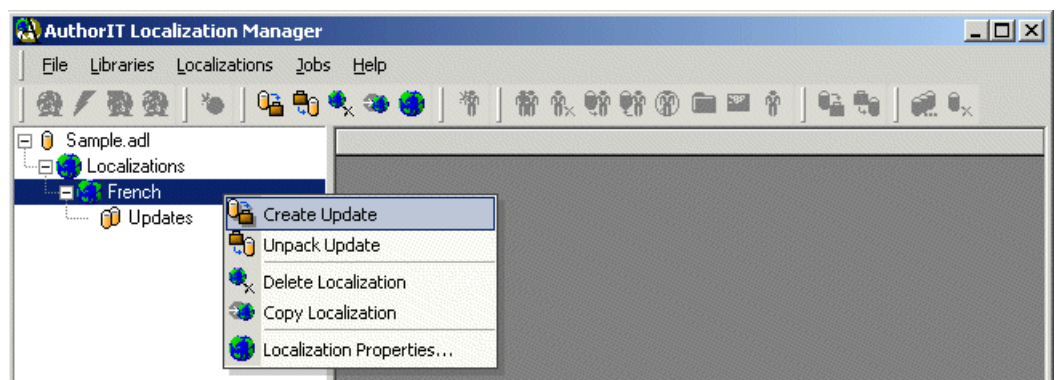
**4)**    Progress is shown as the Target Library is created.
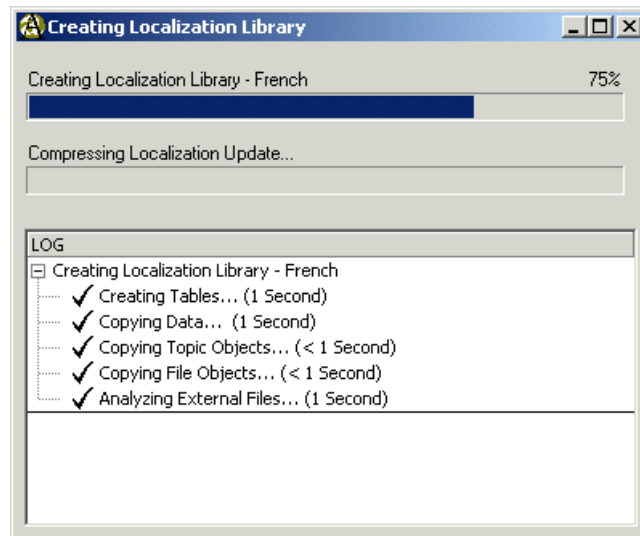


# Creating a Target Library vs Creating a Library Update

The first time you create an update the Target Library is also created.  Two subfolders are created under the folder defined in the *Localization properties* (see "Adding a Localization" on page 19): **Localization Name** and **Localization Name_1** (where 'Localization Name' is the name defined in the Localization properties).
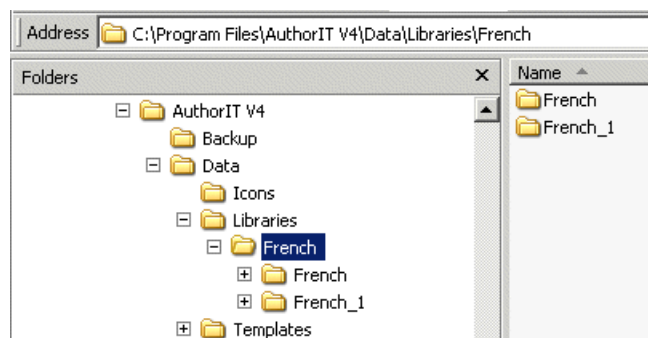


⇥    The **Localization Name** folder contains the **Target Library** that will be translated and used to publish into that localization. Files in this folder (Graphics, Templates, etc) can all be modified for localization but the file names and locations should not be changed.

⇥    The **Localization Name_1** folder contains the first **Localization Update**. Files in this folder should *not* be changed or moved as they are required by the **Compare Update** utility to identify new and modified content.

The Target Library is only created once. Subsequent updates create one additional subfolder for each corresponding update: Localization Name_2, Localization Name_3, and so on.

# Working With Target Libraries Away from the Source Library

There can be circumstances when Target Libraries are administered at a site (or sites) away from the Source Library. For example, you may be sharing the project with a localization vendor and they may be managing the coordination of updates and translation jobs. Or your company may have branches in other countries, and while you work with the source library, each branch may be responsible for their own Localization.
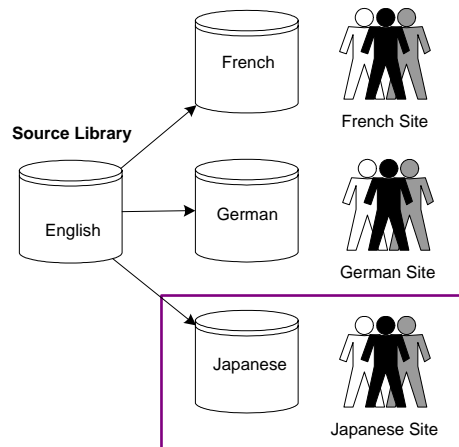


In these circumstances, you will need to create updates from the Source Library and send these to the external site/s. The only file that you need to send is the Localization Name_#.zip file. The Localization Coordinator at the external site then *unpacks the update* (see "Unpacking Updates" on page 48) which extracts the files recreating the folder structure, mirroring that at the main site.

**Note:** The first update *received* at the external site becomes the Target Library.

# Viewing or Modifying Localization Properties

You can view or change the Localization Properties at any time.

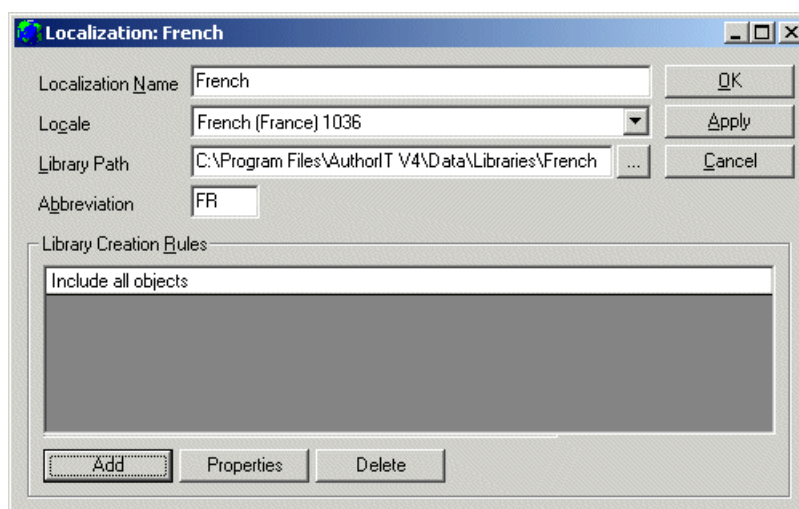## To View or Modify Localization Properties

**1)**    Select the Localization that you want to view or modify

**2)**    Choose the **Localization Properties** button on the main toolbar

-or-

Right click and choose **Localization Properties**

The Properties window for that Localization appears.



**3)**   View and modify the attributes and *object inclusion rules* (on page 21) as
necessary then choose **OK**.

**Note:**  The default Library path sets the name of the subfolder based on what is
first entered in the Localization Name field. If you later rename the Localization,
the path is *not* updated automatically.

# Deleting a Localization

If you no longer use a Localization, you can permanently delete it from the Library.  It
is not possible to permanently delete a Localization if it contains updates as this
would create *orphans*. You must first delete the updates.

**Warning:**  Once you have deleted a Localization, it cannot be recovered. When a
Localization is deleted *all* tracking and historical information linking objects in the
Source Library to the Localization is lost. Be careful.

### *To Delete a Localization*

**1)**   Select the Localization that you want to delete.

**2)**   Choose the **Delete Localization** button on the main toolbar

-or-

Choose **Localizations > Delete Localization** from the main menu

-or-

Right click and choose **Delete Localization**.

**Note:** You cannot delete a Localization if it has updates. The Updates must be
deleted first.

Chapter 4

# Translating the Content

Translation can be carried out directly in the Target Library using AuthorIT or, when the translation is outsourced, the content can be exported as XML and sent to the translators, then imported as XML once translation is complete.

The XML files can be translated with translation memory tools that use XML based editors such as Trados Tag Editor.

We currently have the configuration files available for the following translation memory tools:

❧ *Trados* (http://www.author-it.com/ftp/download/trados_config.zip)

❧ *SDLX* (http://www.author-it.com/ftp/download/sdlx_config.zip)

❧ *DejaVu* (http://www.author-it.com/ftp/download/dejavu_config.zip)

## In This Chapter

# Adding a Translation Job

Translation Jobs are used to define which objects will be exported to XML and sent for translation. Once the Job has been exported as XML, those objects are locked in the Target Library and cannot be modified until they are returned.

A Translation Job has the following attributes:

- ▸▸ **Description:**  A description used to identify the translation job.
- ▸▸ **Translator:**  Used to record the name of the translator the job is assigned to.
- ▸▸ **Export XML to:**  Location of the folder where the XML files will be created.
- ▸▸ **Import XML from:**  Location of the folder where the XML files will be imported from.
- ▸▸ **Export Release State:**  Defines a Release State that is applied to the objects once exported.
- ▸▸ **Import Release State:**  Defines a Release State that is applied to the objects once imported.
- ▸▸ **Status:**  Indicates the current status of the Translation Job.
  - ▸ **Ready for Translation**: The Job has been created but not yet exported as XML.
  - ▸ **Exported for Translation**: The Job has been exported as XML and sent for translation.
  - ▸ **Imported from Translation**: The Job is complete and the XML has been imported.
- ▸▸ **Translation Job Rules:**  A rule, or set of rules, covering which objects will be included for translation.

*To Add a Translation Job:*

**1)**  Select the **Jobs** folder

**2)**  Choose the  button on the main toolbar

-or-

Right click and choose **Add Job...**

**3)**  Enter the Job properties and *creation rules* (see "Translation Job Rules" on page 29).

**4)**  Choose **OK** to add the Job.

# Translation Job Rules

Translation Job Rules control which objects are exported to XML for translation.  For example, you may want to include all objects in a particular Book, or only objects in a specific Release State.

*To Add Translation Job Rules:*

**1)**  Open the Job properties and choose **Add**.

The **Object Inclusion Rule** window appears
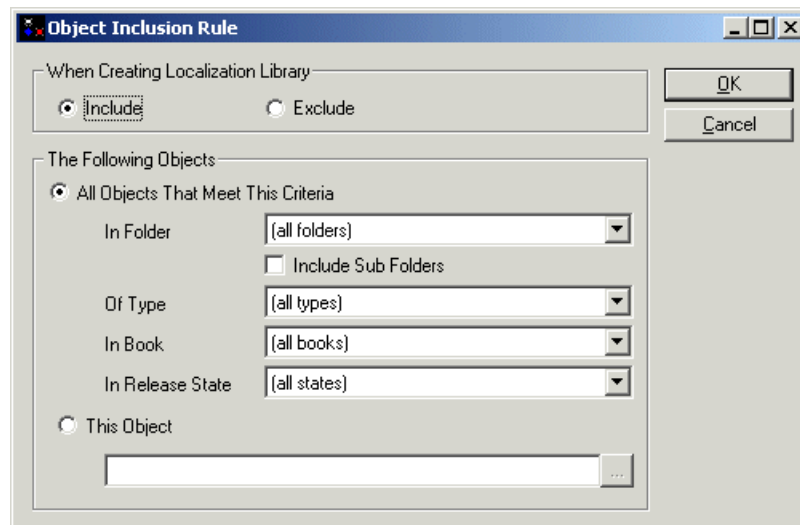


**2)**  Enter your criteria to include or exclude objects and choose **OK**.

**3)**  Select the **Objects** tab to view which objects will be exported to XML for the Translation Job. It is not just Topic objects that are included. Other objects such as the Table of Contents, Index, File and Media objects, also contain translatable content. If these aren't translated, English words will creep into the final document.

**4)**  Repeat these steps to create additional rules.

# Copying a Translation Job

Rather than creating a new Translation Job each time a document requires translation, you can copy an existing Job. When the Job is copied, the same properties and rules are also copied, saving you valuable time. This is ideal when a Book has been modified, and needs to be retranslated.

## To Copy a Translation Job

**1)** Select the Job that you want to copy

**2)** Choose the **Copy Job** button on the main toolbar

-or-

Right click and choose **Copy Job...**

**3)** AuthorIT creates and displays a duplicate of the original Translation Job with "Copy of " inserted at the beginning of the description.

**4)** Change any settings, including modifying the Job Description, then choose **OK** to save your changes.

**5)** A new Job is created with the status of **Ready for Translation**.

# Exporting a Translation Job

Once a Translation Job has been created, it needs to be exported to XML. The XML file can then be sent to the translators.  The XML file can be translated using translation memory tools that support XML based editors such as TRADOS Tag Editor.

## To Export a Translation Job

**1)** Select the Jobs to be exported

**2)** Choose the  button on the main toolbar

-or-

Choose **Jobs > Export Job** from the main menu

-or-

Right click and choose **Export Job**
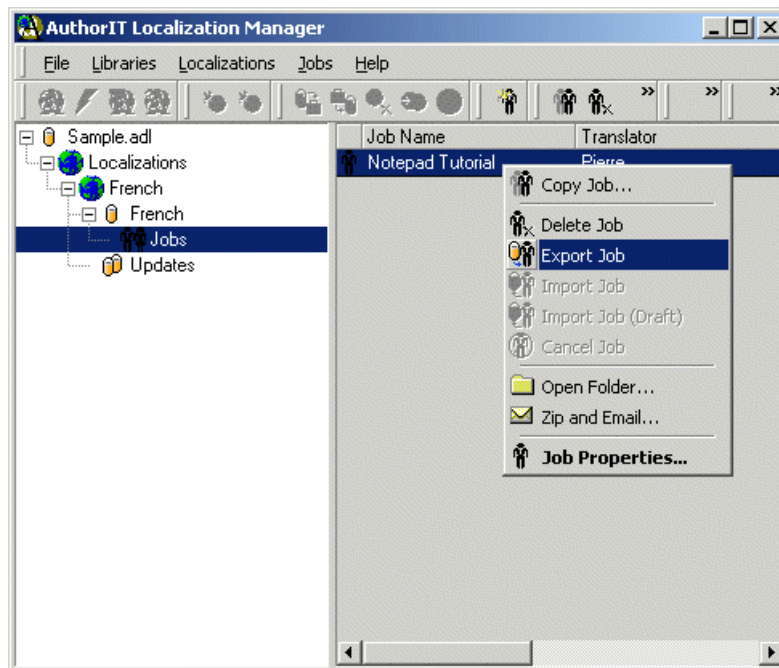
**3)** The Job is exported to XML and the files saved in the folder defined under the Job Properties.

> **Note:** Objects that have been exported are locked in the Target library. This prevents anyone changing the objects while they are being translated.  If an object cannot be locked (for example if it is already locked or the person exporting the Translation Job does not have permissions to modify the object) the object is not exported.

# Translating the XML

Once a Translation Job has been exported as XML, it needs to be sent for translation.

The following files are created during the Job Export. Copies of these should be sent to your Translators:

| File | Purpose |
| --- | --- |
| AuthorIT.xsd | XML Schema Definition file that validates the JobName.xml file. |
| AuthorIT.xslt | XSL Transformation file that allows JobName.xml to be opened in a browser and transformed into something that looks like a document rather than a list of nodes. |
| JobName.xml | This is the exported XML, the file has been named after our Translation Job. |

Translating the XML is done outside of AuthorIT. Your translator will need to be familiar with translating XML using a translation memory or other tool that can translate XML files (such as TRADOS Tag Editor).

**Note:** To view an exported job (XML file) you will need IE6 or another browser that supports the W3C recommendations.
IE5 and IE5.5 do not interpret the XSLT correctly as they were released with only a "working draft" of the W3C XSL recommendation. The specification changed before the "official" release but IE5/IE5.5 didn't.
IE6 will display the XML files perfectly and is compliant with the W3C XSL recommendation.

# Sending the XML to Your Translator

Once a Translation Job has been exported, it needs to be sent to your Translators. Provided you have a MAPI compliant email client installed, Localization Manager can zip the files and begin creating a new email message for you.

### *To Send the XML Files to a Translator*

**1)** Select the Exported Job

**2)** Choose the **Zip and Email** button on the main toolbar

-or-

Choose **Jobs > Zip and Email...** from the main menu

-or-

Right click and choose **Zip and Email...**
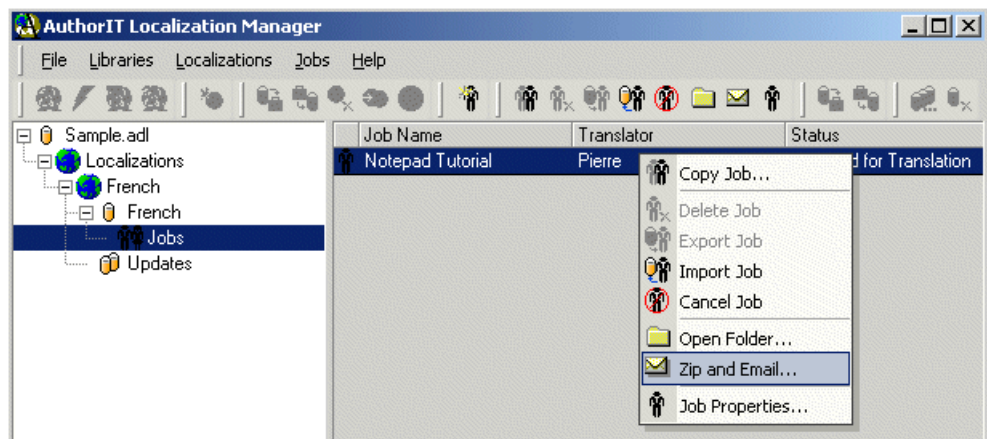


**Note:** This option is only available when the Job is in the status of **Exported for Translation** and the XML Files have been created.

**3)**   A new email is created with the zipped XML files attached. The Translator name defined in the Job Properties is used in the email **To...** field. The Description defined in the Job Properties is used as the email **Subject**.

**Tip**: Add your Translator to your Address Book so Outlook can automatically check and detect their email address.   Alternatively, use their email address in the Translator field when creating the Job.
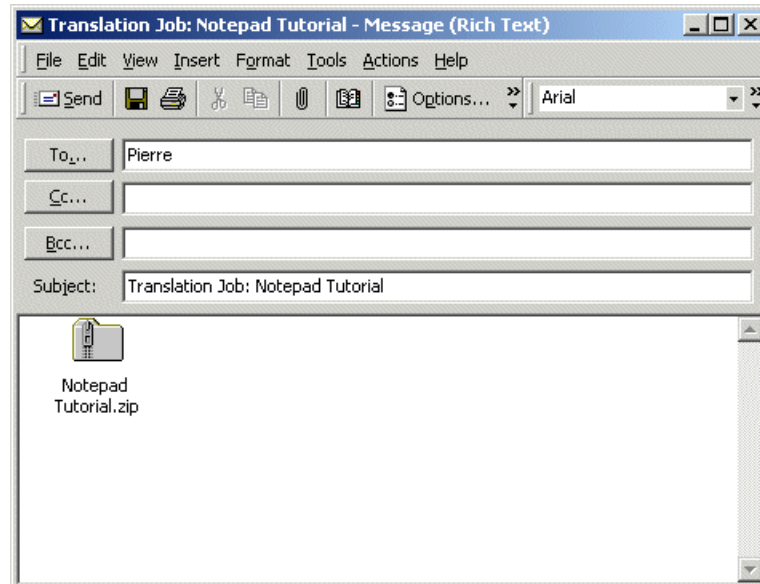


You may also want to include a *context document* (see "Publishing a Context Document for Translation or Review" on page 34) to assist your translators in determining context.

# Publishing a Context Document for Translation or Review

To assist the translation and/or review process, it's a good idea to include a published Word or PDF document. Rather than just seeing a couple of topics, sending them the whole document means they can see the big picture. In addition to this, all objects that are in a release state where a review color has been defined, appear highlighted in the Word/PDF document making it easier for a translator to determine context, or for a reviewer to identify what content should be reviewed. This is particularly helpful when:

⏵ A Book needs to be retranslated after being added to or modified

⏵ Topics are reused in multiple Books so have already been translated

**Warning:** If including a Word document with the XML files for translation, it's important that your translators are aware that the Word document is for reference purposes only and it is the XML that requires translation. We recommend sending a PDF so your translators aren't tempted to carry out the translation in Word.

## *To Publish a Context Document:*

**1)** Select the Target library containing the document you want to publish

**2)** Choose the **Open Library with Publisher** button on the main toolbar

-or-

Choose **Libraries > Open Library with Publisher...** from the main menu

-or-

Right click and choose **Open Library with Publisher...**

The Library opens with the AuthorIT Publisher

**3)** Select **Publish**, choose the relevant Book(s), and click **Start**
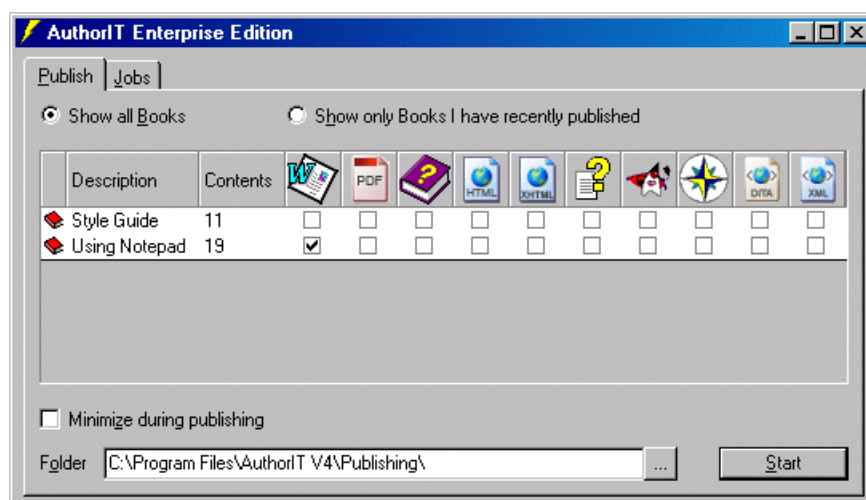


**4)** The Publishing Log displays a list of objects that are not yet localized.

Once an object is translated, it is flagged as **localized**.
While the rules defined in a translation job determine which objects are included in the exported XML, objects that are flagged as localized are not included.
When an update is created and a modified object is accepted, this flag is removed until it has been translated again.



> (i) Object 'To print a Notepad document' (177) not localized.
> (i) Object 'To insert the time and date in a document' (178) not localized.
> (i) Object 'To append a log to a document' (179) not localized.
> (i) Object 'Using different language formats in Notepad' (180) not localized.
> (i) Object 'Chapter Section' (36) not localized.
> (i) Object 'Index Section' (41) not localized.
> ☑ ✓ Resolving References and Footnotes... (< 1 Second)
> ✓ Applying Styles... (6 Seconds)
> ✓ Updating Fields ... (13 Seconds)
> ✓ Creating Table of Contents and Index... (< 1 Second)
> ☐ ✓ Completing Glossaries... (< 1 Second)
> ⚠ This book contained 44 Objects that aren't localized.

**5)**   The published document has highlighting to indicate the translatable content.
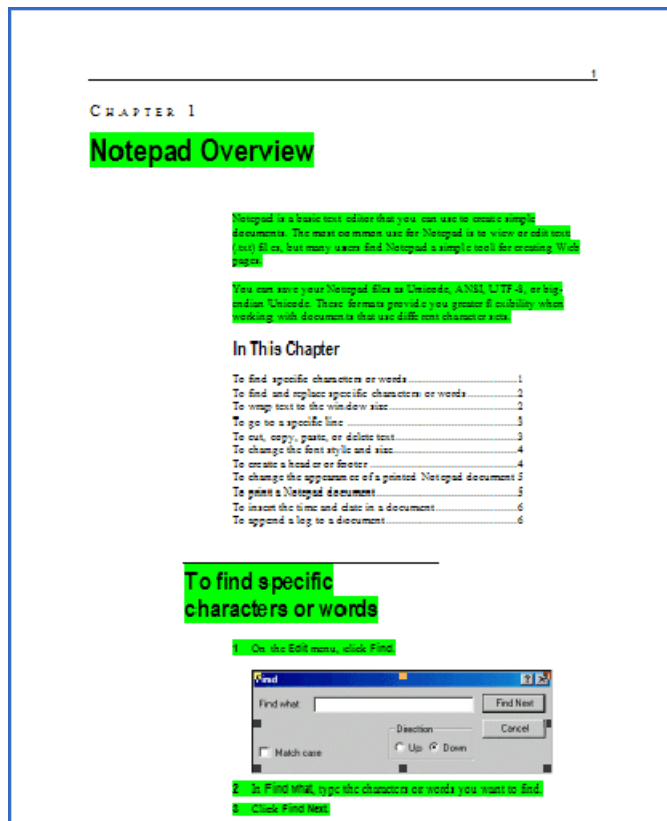
# Localization of Variables

Variables are used to substitute text at the time of publishing. They can be used in many publishable fields such as Book or Topic descriptions, Topic text and Hypertext links. They work by looking for specific text (for example **<PRODUCT>**) and replacing it with a variable assignment (for example **Notepad**).

When a translation job is exported to XML, any variable assignments on Book and Topic objects are exported as well. To a translator these show up as part of the Topic or Book they belong to.

Translators need to be aware that text enclosed in <angle brackets> may have special meaning and should be provided with rules to deal with each situation.

Because of the many ways that variables can be used, they will generally be dealt in one of three ways:

**1)   Variable assignment is translated.**

The variables themselves should not be translated, but left 'untouched'. Only the variable assignments should be translated. Otherwise the variable cannot be resolved during publishing, and the final output will display the translated variable between <angle brackets>.

**Note:** It will often be appropriate to move the location of the variables due to different word orders.

## Topic - To append a log to a document (179)

| Object | Description | To append a log to a document |
| | GUID | 918ACEC7723B41CAA9978EC940BA1D62 |
| | ID | 179 |
| Headings | All | To append a log to a document |
| Text | | On the first line of a <PRODUCT> document, type the following extension at the left margin, making sure you use all uppercase letters and include the period: |

.LOG

> Variable displayed within Topic text. This **should not** be translated.

On the **File** menu, click **Save**.

**Note:** Every time you open this document, <PRODUCT> will append the current time and date to the end of the document, as specified by your computer's clock.

| VariableAssignments | VariableAssignment | ID | 19 |
| | | Value | Notepad |
| | | ValueObject | 0 |
| | | Style | -1 |

> Variable assignment value. This **should** be translated.

**2)   Variable default value is translated**

Variables do not always have assignments against them. In these circumstances, the default value (as defined in AuthorIT Administrator) is used.
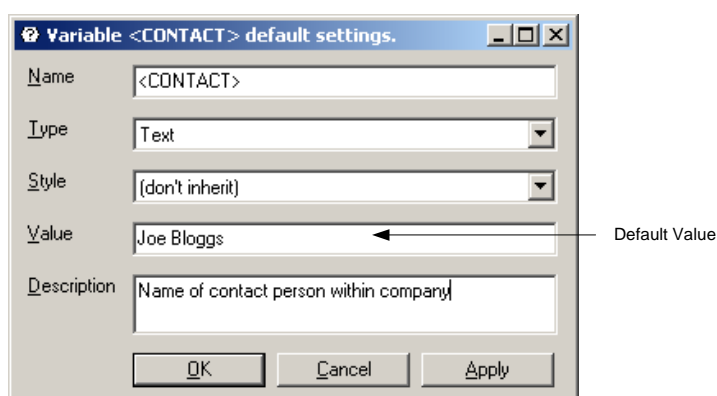


The default value is not exported to XML and, where necessary, should be translated within the Target library.

**3)   Variable is removed from text.**

In some circumstances the variable should be removed and replaced with regular text.

For example, let's say we have defined a variable **<COLOR>** where we substitute either the word "color" (for the US market) or "colour" (for the UK market). Neither of these make much sense in the French or German market, and while we could assign a new value, it's probably easier to replace the variable with regular text - replacing <COLOR> with `couleur` in French, and with `Farbe` in German.

# Localizing Templates and Objects Directly in AuthorIT

Provided you have a good working knowledge of AuthorIT and a good understanding of the language you are translating into, there are some objects that are easily translated directly in AuthorIT.

These changes should be made in the Target Library, *not* your Source Library.

For example, let's look at the Hypertext Link object **Jump Template**. It contains the text `see` and `on page`, which is then used by all Hypertext Link objects based on that template. That text won't make much sense to a non-English audience, so needs to be translated.

## *To Translate an Object Directly in the Target Library*

**1)**   Open the Target Library in AuthorIT.

**2)**   Locate and open the object you want to localize.

**3)**    Translate the publishable content from English text...



...into localized text...



**4)**    Select **OK** to save the object.

> **Note:** Many non-topic object properties that need translation are inherited from the template they are based on and cannot be changed directly. To change these properties, you'll need to edit the object template. Changing the template, will update all objects based on it.

Once an object is translated, it needs to be flagged as **localized**. While the rules defined in a translation job determine which objects are included in the exported XML, objects that are flagged as localized are not included. When an update is created and a modified object is accepted, this flag is removed until it has been retranslated.

This flag is set automatically when an object is translated via exporting and importing the XML. When localizing an object directly in the Target Library, you need to tell AuthorIT that it has been translated by flagging it as localized.

## To Flag an Object as Localized:

**1)**    Select the object

**2)**   Right click and choose **Localized**
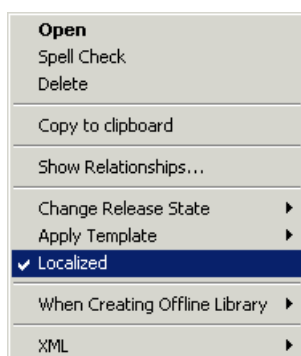


# Glossary Objects

Glossary objects should have the following properties translated:

| Tab | Field |
| --- | --- |
| Document | Glossary Title |
| HTML | Glossary Title |

These fields control the heading at the beginning of a Glossary.

# Hypertext Link Objects

Hypertext Link objects should have the following fields translated:

| Tab | Field |
| --- | --- |
| Document | Reference Text<br> - and -<br>Page Number text |

These fields control the appearance of the cross reference in the printed output.

# Index Objects

Index objects should have the following properties translated:

| Tab | Field |
| --- | --- |
| Document | Index Title |
| HTML | Index Title |

These fields control the heading at the beginning of an Index.

## Media Objects

Media objects don't contain *translatable* properties. However, the paper size may vary across countries. These properties need to be Localized.

| Tab | Field |
| --- | --- |
| Document > Paper Size | Page Size |
| Document > Paper Size | Height |
| Document > Paper Size | Width |

Usually this will be A4 or Letter.

## Table of Contents

Table of Contents objects should have the following properties translated:

| Tab | Field |
| --- | --- |
| Document | TOC Title |
| WinHelp | Contents Title |
| HTML | Contents Title |

These fields control the heading at the beginning of a Table of Contents.

## Table of Figures

Table of Figures objects should have the following properties translated:

| Tab | Field |
| --- | --- |
| Document | TOF Title |

This field controls the heading at the beginning of a Table of Figures.

## Topic Objects

It is obvious that Topics need to have their Headings and Text translated. However, it could be easy to miss the Related Group Headings. Typically, these are usually inherited from whichever Topic Template the Topic is based on.

Topic Object Templates (and Topics containing Related Groups) should have the following properties translated:

| Tab | Field |
| --- | --- |
| Related Group(s) | Group Heading |

These fields control the headings above the related topic groups created during publishing. For example, "See Also", "Next", "Previous" and "In this Chapter".

# Importing a Translation Job

Once translation is complete, you need to import the translated XML files.

**Note:** If translation is only partially complete, or the translated file is being reviewed externally, you should choose to import a *draft translation job* (see "Importing a Draft Translation Job" on page 43). This updates the objects in the target library, allowing you to publish the updated content but leaves them locked until the job is imported again.
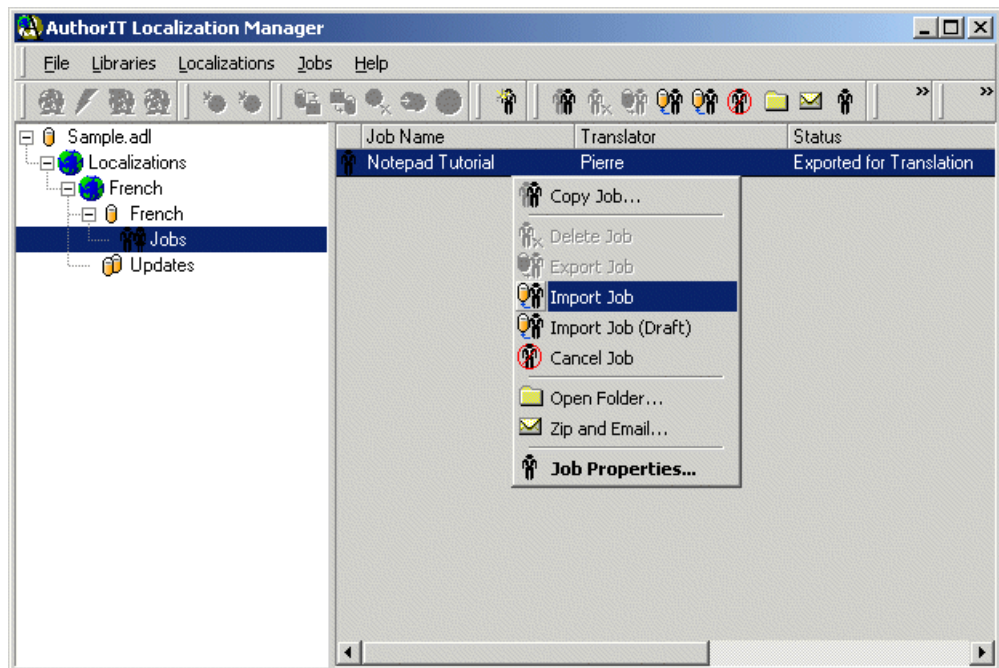
### *To Import a Translation Job:*

1) Copy the translated XML files to the folder defined in the **Import XML from** field under the Translation Job properties.

2) Open Localization Manager and select the Job to be imported

3) Choose the ![button icon] button on the main toolbar

   -or-

   Choose **Jobs > Import Job** from the main menu

   -or-

Right click and choose **Import Job**.



**4)**   The objects in that Translation Job are imported and unlocked in the Target Library.

# Importing a Draft Translation Job

You can choose to import a "draft" of a translation job. This updates the objects in the target library, allowing you to publish the updated content but leaves them locked until the job is imported again. The status of the job remains as **Exported for Translation** allowing the job to be re-imported at a later stage.  This is ideal if translation is partially complete, or the translated file is being reviewed externally.
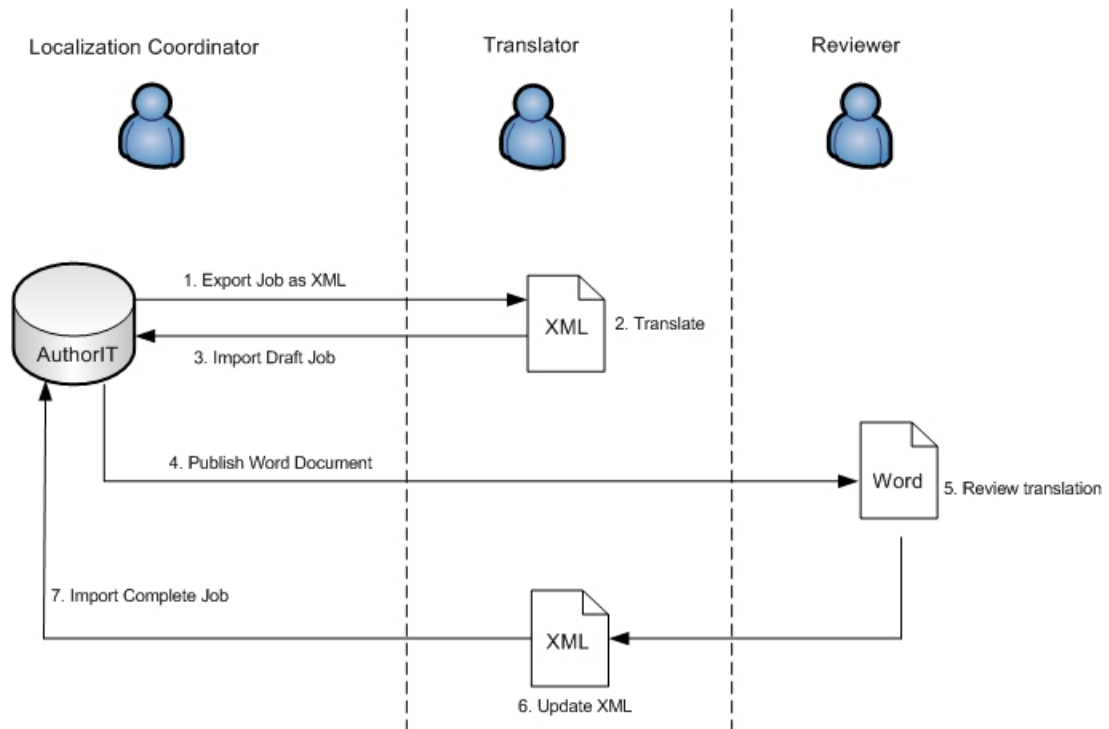


1. Localization Coordinator creates translation job and sends XML to translator
2. Translator translates XML file and returns to Localization Coordinator
3. Localization Coordinator imports XML file as a "Draft" job
4. Localization Coordinator publishes Word document and sends to reviewer
5. Reviewer adds comments/corrections and sends back to translator
6. Translator updates XML file and returns to Localization Coordinator
7. Localization Coordinator imports translation job.

## *To Import a Draft Translation Job:*

**1)**   Copy the translated XML files to the folder defined in the **Import XML from** field under the Translation Job properties.

**2)**   Open Localization Manager and select the Job to be imported

**3)**   Choose the 🔲 button on the main toolbar

-or-

Choose **Jobs > Import Job (Draft)** from the main menu

-or-

Right click and choose **Import Job (Draft)**.

**4)**    The objects in that Translation Job are imported but remain locked in the Target Library.

# Viewing or Modifying Translation Job Properties

You can view or change the properties of a Translation Job at any time.

## *To View or Modify Translation Job Properties*

**1)**  Select the Translation Job that you want to view or modify

**2)**  Choose the **Job Properties** button on the main toolbar

-or-

Choose **Jobs > Job Properties...** from the main menu

-or-

Right click and choose **Job Properties...**

The Properties window for that Translation Job appears.

**3)**  View and modify the attributes and *Translation Job Rules* (on page 29) as necessary then choose OK.

# Cancelling a Translation Job

There can be circumstances when you need to cancel a Translation Job. For example, you may have defined the rules incorrectly and the wrong information was included when it was exported. Or content in the source library may have been updated after the job was exported but before the initial translation could be started. If the initial translation has not begun, cancelling the job allows you to Export the Job again with the updated information.

**Note:** If you've already sent the XML to your Translators, remember to let them know that the Job has been cancelled so they don't inadvertently translate the old files.

## *To Cancel a Translation Job*

**1)**  Select the Translation Job that you want to cancel

**2)**  Choose the **Cancel Job** button on the main toolbar

-or-

Choose **Jobs > Cancel Job** from the main menu

-or-

Right click and choose **Cancel Job**

A message appears asking you to confirm the action

**3)**  Choose **Yes** to cancel the job. The object locks that were created when the job was exported are removed in the Target library allowing you to modify the job properties and export it again.

**Note:** The XML file that was created when the job was first exported will need to be deleted manually before the job can be exported again. Likewise, the release state that was applied to the objects in the Target Library when they were first exported is not removed automatically and will need to be modified manually if necessary.

# Deleting a Translation Job

If you've created a Translation Job in error or wish to remove an old Job that is no longer needed, it can be deleted permanently from the Target library.

**Warning:** Once you have deleted a Translation Job, all record of that Job is removed from the Target library. You will not be able to import any XML that may have been exported before the Job was deleted. All tracking records will be lost. Be careful.

## To Delete a Translation Job

1) Select the Translation Job that you want to delete

2) Choose the **Delete Job** button on the main toolbar

   -or-

   Choose **Jobs > Delete Job** from the main menu

   -or-

   Right click and choose **Delete Job**

   A message appears asking you to confirm the action

3) Choose **Yes** to delete the Job. If the Job had been exported, any object locks that were created are removed in the Target library.

   **Note:** Any Release State that was applied to the objects in the Target library when they were first exported is not removed automatically and will need to be modified manually if necessary.

Chapter 5

# Managing Updates to Content

Translation is not normally a one off process. Any update to the content in the Source library will normally require retranslation in the Target library.

## In This Chapter

## Creating Library Updates

From time to time it will be necessary to update Target libraries to ensure that they get the benefits of any new content and modifications to the existing content.

Localization Manager creates a **Library Update** from the Source Library which is then compared to the Target Library using the **Compare Update** utility. New or modified content is identified, and can be accepted or declined by the Localization Coordinator. The content is then synchronized in the Target Library.
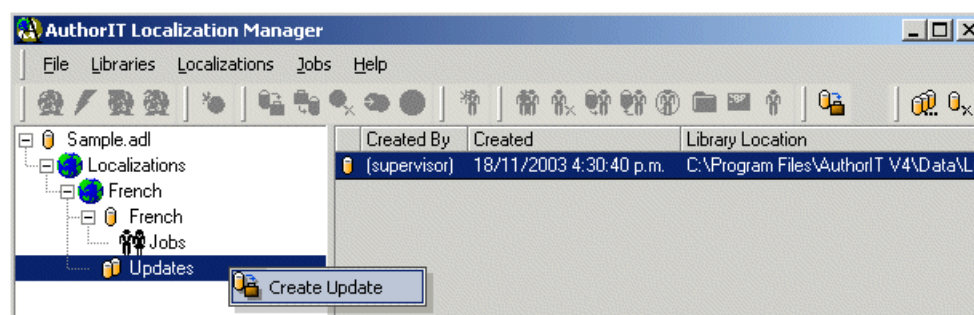
### To Create a Library Update:

**1)** Open the Source Library with Localization Manager

**2)** Select the **Updates** folder for the corresponding Localization.

**3)** Choose the ![button] button on the main toolbar

-or-

Choose **Localizations > Create Update** from the main menu

-or-

Right-click and choose **Create Update**

**4)** A subfolder is created under the folder defined in the *Localization properties* (see "Adding a Localization" on page 19), containing the **Localization Update**. Files in this folder should *not* be changed or moved. These files act as a snapshot of the Source library at the point the Update is created and are used by the Compare Update utility to identify new and modified content between updates.

Each update creates its own folder. For example French_2, French_3, and so on.

**5)** If the Target Library is being administered at an external site (for example, if you are working with a localization vendor, or have multiple branches each responsible for their own Localization), you'll need to send them the zipped update file found in the new update folder. They can then *unpack the update* (see "Unpacking Updates" on page 48), and use the Compare Update utility to synchronize contents.

# Unpacking Updates

Each update file (named **Localization Name_#.zip**) contains an AuthorIT library and the accompanying external files. A new update file is created each time the **Create Update** function is run. This file is unpacked automatically when working in the Source Library.

When you don't have access to the Source Library and only work with a Target Library, you will be sent the (zipped) update file. You'll then need to unpack the update which extracts the files and recreates the folder structure.

The process is slightly different for the first update received, as it also becomes the Target Library.

## *To Unpack the First Update:*

**1)** Start the Localization Manager.

**2)** When the Open Library window appears, choose **Cancel**.
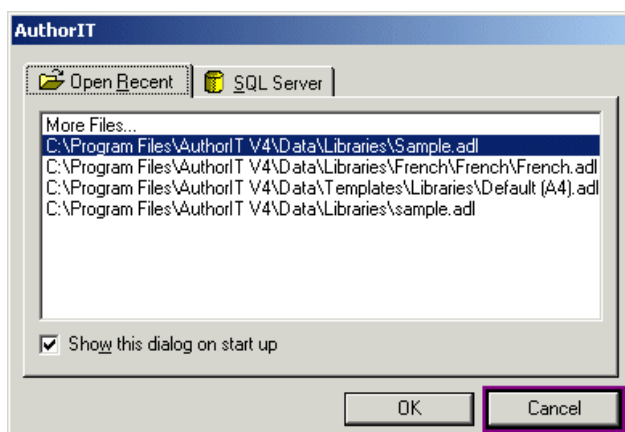


**3)** Choose **File > Unpack Update...** from the main menu. This option is not available when a library is open.

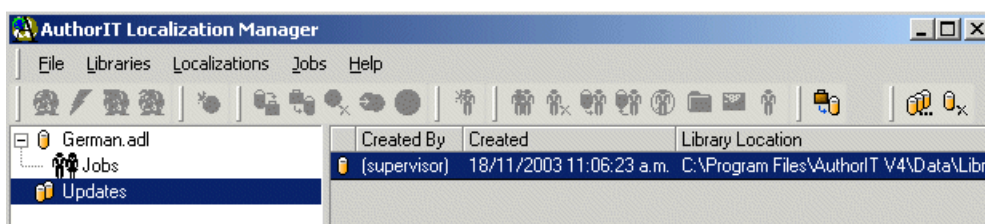The Select a File window appears

**4)** Locate and select the zipped update file, then choose **Open**.

The Target Library is created and the first update is extracted. The *folder structure* (see "Creating a Target Library vs Creating a Library Update" on page 24) is created relative to the Library path defined in AuthorIT under **Tools > Options > File Locations**.

As additional updates are created and received, these must also be unpacked. The process is slightly different, as the Target Library has already been created.

### To Unpack Subsequent Updates:

1)  Open the Target Library

2)  Select the Updates folder



3)  Choose the [icon] button on the main toolbar

    -or-

    Right click and choose **Unpack Update**

    The Select a File window appears

4)  Locate and select the zipped update file, then choose **Open**

    The Update is extracted.

# Using the Compare Update Utility

When a Library Update is created, the Compare Update utility indicates which content is new and which content has changed.  The Localization Coordinator then accepts or rejects these objects for synchronization between the Source and Target libraries.

### To Use the Compare Update Utility:

1)  Select the Update you want to compare

2)  Choose the [icon] button on the main toolbar

    -or-

Right-click and choose **Compare Update**.
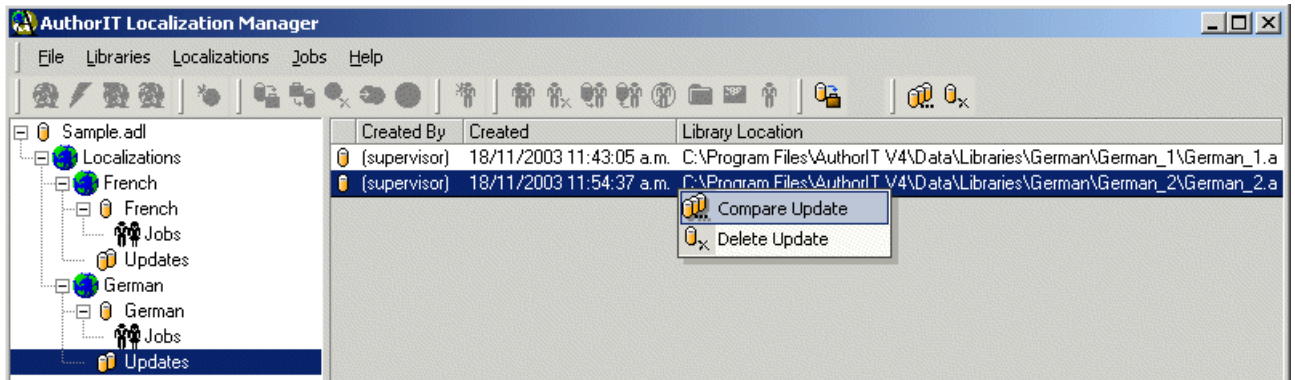


**3)** The Library Compare window appears, displaying the differences between the **Update**, the **Previous Update**, and the **Target**. New and modified Objects are shown in bold for quick and easy identification. Objects that have been modified since the last update are also highlighted with a yellow background.
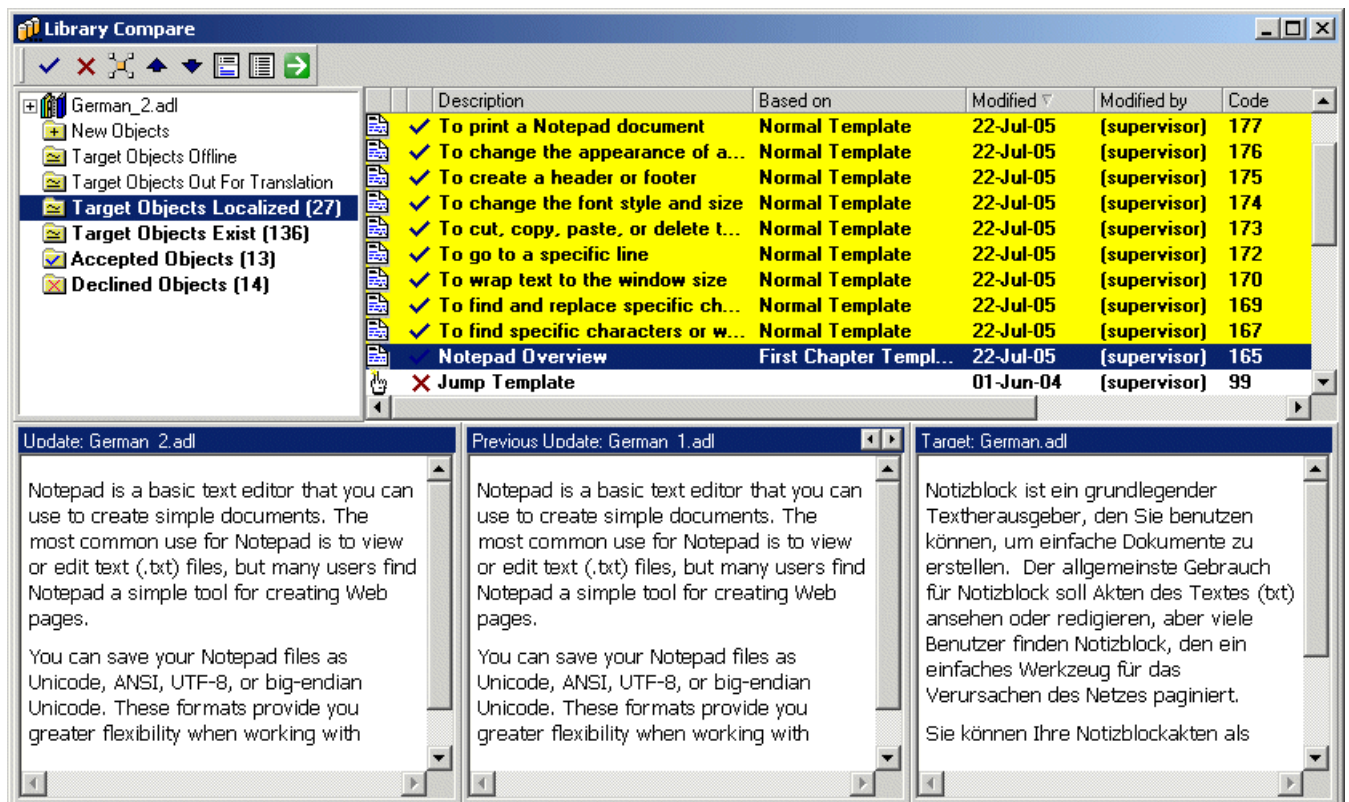


All objects (and their dependencies) appear in *one* of the following special folders.

‣ **New Objects** - lists all objects in the update that are new to the target library. This includes newly created objects, and objects that had previously not been been made available in the locale but have now been included due to a change to the Localization Rules. All objects in this folder are automatically marked as Accepted as they do not exist in the target locale.

**Note:** Declining any of these objects is likely to result in the update not being able to continue due to dependencies not being included.

‣ **Target Objects Offline** - lists all objects that are part of the update that have been checked out of the target library to an offline library. Unless you are creating offline libraries directly from your target libraries this folder should always be empty. All objects in this folder are automatically marked as Declined as they cannot be overwritten while the target objects are locked.

‣ **Target Objects Out For Translation** - lists all objects that are part of the update that are already in the process of being translated in the target library and have been sent out in a translation job. All objects in this folder are automatically marked as Declined so they don't overwrite objects already in the process of translation. You can override this by selecting the object and accepting it, but this should only be done where a change to the source object is critical and requires retranslation. Any objects accepted will be overwritten with the source and will prevent that object from being imported with the original job it was exported in.

‣ **Target Objects Localized** - lists all objects that are part of the update that are already translated/localized in the target library. All objects in this folder are automatically marked as Declined so they don't overwrite objects already translated. You can override this by selecting the objects you need to retranslate and accepting them. This should only be done when you are actually updating a previous translation of an object. Any objects accepted are overwritten with the source and will require retranslation.

‣ **Target Objects Exist** - lists all objects that are part of the update that already exist in the target library but have not yet been localized. Any object listed here does not appear in any of the other special folders. All objects that have been modified since the previous update are automatically marked as Accepted as they have changed. We are not able to determine the importance or significance of the change, just that a change has occurred. It is reasonably safe to accept these as the target objects are not yet localized.

**Important:** If someone manually deletes the previous update library, Localization Manager  will not be able to determine what has changed as there is nothing to make the comparison with.

4) Select an object, and it displays in the three preview windows where the differences can be compared. You can alternate between **Normal Preview** 📑 and **Property Sheet** 📄 views.

5) To accept an object, select it and choose the **Accept Change** ✔ button

There can be instances when you don't want an object that has been changed in the Source library to be updated in the Target library.  For example a contact telephone number of the local US branch may have changed, while the French number remains the same.

6) To decline an object, select it and choose the **Decline Change** ✖ button

💡 Tip: Use the SHIFT or CTRL keys to select and accept or decline multiple objects at a time.

7) When ready, choose the **Import Accepted Objects** ➡ button.

A message appears asking you to confirm the action

8)  Choose **Yes** to import the accepted changes

Progress is shown as the accepted changes are imported and the Target library updated.

# Deleting an Update

If you no longer use an Update, it can be deleted. However, remember that an update is the only means of determining when new content has been introduced, or when content has been changed. Unless you are trying to save disk space, there is normally no need to delete it. If disk space is an issue, we suggest that rather than deleting the Update, you remove the physical update files but retain the folder structure and zipped update file. The Zip file can then be extracted or unpacked at a later stage for comparison if needed.
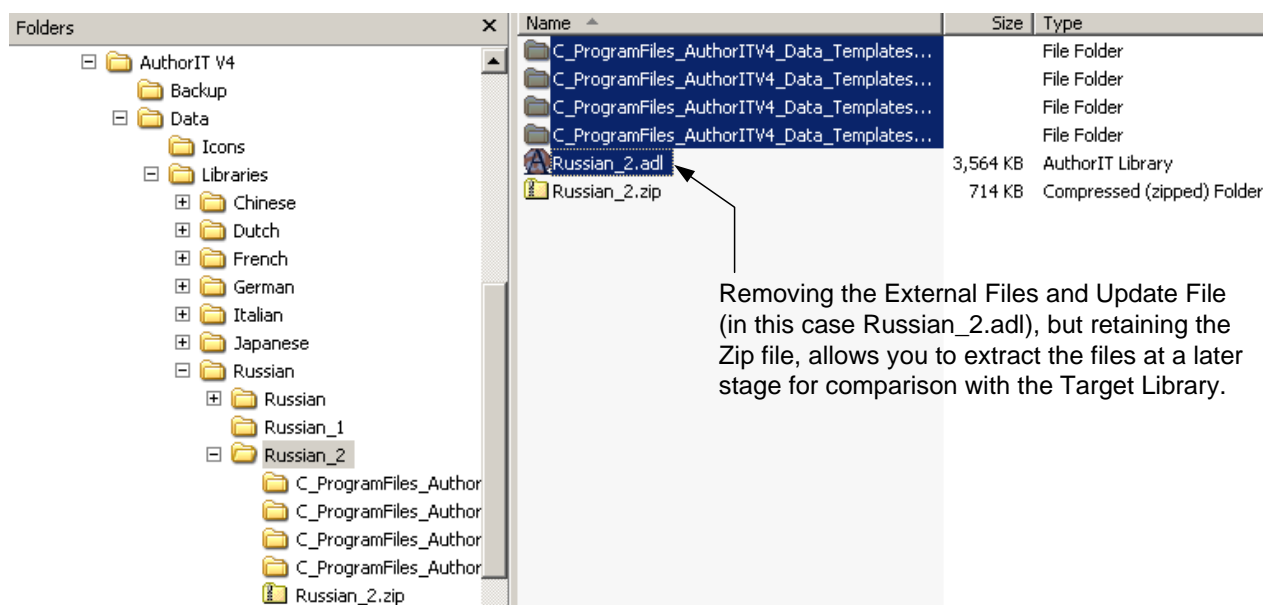


Removing the External Files and Update File (in this case Russian_2.adl), but retaining the Zip file, allows you to extract the files at a later stage for comparison with the Target Library.

## *To Delete an Update:*

1)  Select the Update that you want to delete.

2)  Choose the **Delete Update** button on the main toolbar

-or-

Right click and choose **Delete Update**.

A message appears asking you to confirm the action

3)  Choose **Yes** to delete the Update.

---

**Note:** Only the relationship between the Target library and the Update is deleted. The physical files still exist, and need to be removed manually via Windows Explorer.

# Publishing From a Target Library

Publishing from a Target library is *almost* the same as publishing from any other library, with a couple of small exceptions:

▸ **External File Locations**

Files external to your Library (such as Word templates, HTML templates and linked graphics) are stored in sub folders under your Localization folder. These folders have names that are an abbreviated version of where the file was originally stored. The path to this external file that you see inside an object is not the location that the Publisher will actually look for it.

▸ **Publishing Folder**

The folder your published output is created in will now also contain the name of your Localization. This is to prevent the English version overwriting the French (or vice versa) when you publish.

Most importantly, however, you must ensure the correct language settings are used on the machine you are publishing from. This involves setting the **system locale** (see "What is the System Locale?" on page 54) to the desired language and using the correct **input locale** (see "What are Input Locales?" on page 54) before publishing.

## *To Publish Directly from Localization Manager:*

**1)**   Select the Target library that you wish to publish from

**2)**   Choose the **Open Library with Publisher** button on the main toolbar

-or-

Choose **Libraries > Open Library with Publisher...** from the main menu

-or-

Right click and choose **Open Library with Publisher...**

The Library opens with the AuthorIT Publisher

**3)**   Select **Publish** and choose the Book/s and relevant outputs you wish to publish to.

**Note:** You can also publish from the Source library in this way.

# What is the System Locale?

The system locale (sometimes referred to as the system default locale), determines which codepages and associated bitmap font files are used as defaults for the system. A codepage is a character set that can include numbers, punctuation marks, and other glyphs. These codepages and fonts enable applications to run as they would on a system localized to the language of the system locale.

The system locale is defined when installing Windows, but can later be modified in the Regional Settings Control Panel.

**Note:** Changing the system locale requires a reboot and only those system locales for which appropriate *language groups* (see "What is a language group?" on page 54) have been installed will be available to choose from.

# What are Input Locales?

Input locales are pairings of an input language with an input method (which might be a particular keyboard layout, an Input Method Editor, or speech-to-text converter, for example). Specifically, an input locale describes the language being entered, and how it is being entered.

Input locales are added on a per user basis. For each account it is possible to install multiple input locales and switch between them when entering text, allowing for the composition of multilanguage documents.

**Note:** Adding or removing input locales has an immediate effect (ie. a reboot is not required). Only those input locales for which appropriate *language groups* (see "What is a language group?" on page 54) have been installed will be available to choose from.

# What is a language group?

A language group is a set of all the keyboard layouts, IMEs, fonts (TrueType or bitmap), font links, language packs (LPKs) and codepage translation tables needed by the system to support the given group of languages. The list of installed language groups controls which input and system locales can be selected.

Installation of a language group copies (but does not activate) the necessary keyboard files, Input Method Editors (IMEs), TrueType font files, bitmap font files, and National Language Support (.nls) files to the system. Adding a language group also adds registry values for Font Linking and installs scripting engines for Complex Script languages (Arabic, Hebrew, Indic, and Thai).

# Glossary of Terms

## C

### Context Document

A published Word or PDF document, used to assist the translation and/or review process. All objects that are in a release state where a review color has been defined, appear highlighted in the Word/PDF document making it easier for a translator to determine context, or for a reviewer to identify what content should be reviewed.

## E

### Exact Match

The occurrence of a sentence or phrase in a file that is identical to a sentence or phrase stored in a translation memory.

## F

### Fuzzy Match

When a sentence or phrase in a translation memory is similar (but not identical) to the sentence or phrase the translator is currently translating.

## L

### Library Update

A zipped file created by Localization Manager that contains an AuthorIT library and accompanying external files. This is used as a *snapshot* of the objects at the point the Update is made and is used by the Update Compare utility to display and import new and modified objects from the Source Library.

### Localization

Usually, a country or language that requires the documentation to be translated. However, this could also be variations on the same language such as English (US) and English (UK) or clients that require wildly different versions of the same base library.

### Localization Coordinator

The person who uses the AuthorIT Localization Manager program to coordinate and manage the localization process.

### Localization Manager

An AuthorIT module that creates, manages and updates Target Libraries and exports and imports translation jobs as XML.

### Localize

The process whereby objects are converted from one localization to another. This often involves translation into another language but also includes non-translation aspects such as paper size and fonts/styles. This can be accomplished internally (within the AuthorIT software) or externally (by XML export/import to software such as TRADOS).

## S

### Source Library

The original (or main) library containing content in the starting language.

## T

### Target Library

The *live* library that eventually contains the translated/localized content. It is from this library that the localized content is published.

### Translation Memory

A database of sentences and phrases and their translated equivalent, built from previous translations. Translation memory tools increase efficiency by allowing the reuse of previously translated phrases and sentences. Popular translation memory tools include TRADOS, Deja Vu, Star Transit, and SDLX.

# Index