

Internationalization Guidelines for Software Development

Pearson Globalization Services

Contents

Why is Internationalization Needed?	3
What's the Difference between Translation, Localization (L10n) and Globalization (G11n)?	3
What is Internationalization (i18n)?	3
Software Localization Project Components	4
Typical Software Localization Project Workflow	5
Internationalization of the User Interface (UI)	7
Regional Standards.....	7
Checklist for Translatable Files	8
Checklist for UI Design	9
Potential Issues if Concatenation is Not Handled Properly	10
Checklist for Locale Support.....	10
Language Use in Software Projects	11
Project Glossary	11
Pearson Translation Request Form	11
Continuous Translation.....	11
Testing and Quality Assurance (QA).....	12
Appendix A — Language Codes and Regions	13
Appendix B — Localization Definitions	14
Contact Us for More Information.....	17
Join us on Neo	17

Internationalization Guidelines Across Pearson

More and more, Pearson looks to new markets across the globe to expand its educational reach. With this eye on global markets, the importance of internationalization becomes paramount. This guide serves as an overview of some best practices in internationalization for Software Development.

Why is Internationalization Needed?

The localization of a product does not begin with the delivery of source material to the translation vendor. Software developers can influence the success of their localization and globalization effort to a large extent by preparing their products for foreign markets during the development phase, in other words, by *internationalizing* them properly.

A global product needs to be functional and accepted in international markets; the product should be designed to support features such as international character sets, keyboard layouts, date and time formats, and currencies. The user assistance and documentation should be concise, clear, and not contain any jargon or slang, and its contents should be free of culture-specific examples or references.

Ensuring that a product is localizable will help to reduce costs by developing the product in a way that ensures a smooth localization process, both for the software developer and the localization vendor. One way to do this in software is to ***externalize all translatable components into resource files*** to make them easy to identify and translate, and avoid hard-coding. In user assistance and documentation, using generic country-independent examples or creating multi-layered images reduces costs considerably.

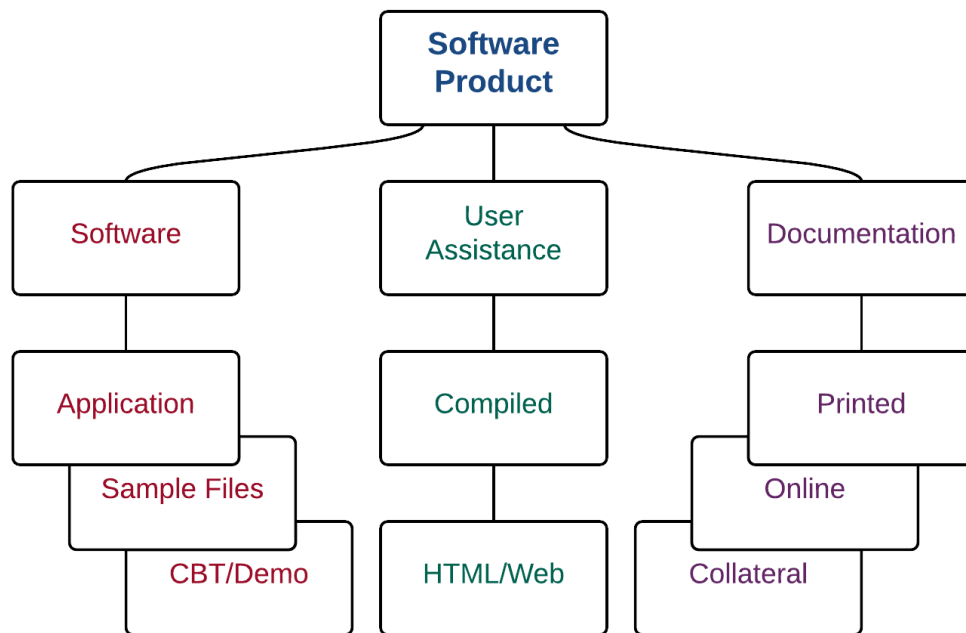
What's the Difference between Translation, Localization (L10n) and Globalization (G11n)?

WHAT IS LOCALIZATION?	LOCALIZATION vs. TRANSLATION?	WHAT ARE THE BENEFITS?
Localization (L10n) is the process of adapting products and services to fit <i>linguistic, cultural, technical and other locale-specific requirements</i> such as date, time formats and numeric formats.	Localization differs from translation because it involves a comprehensive study of the target culture in order to <i>correctly adapt the product to local needs.</i>	Localization and cultural customization can lead to creating <i>effective global products.</i> Globalization (G11n) is the integration of these concepts, <i>from design to deployment.</i>

What is Internationalization (i18n)?

Internationalization is the process of generalizing a product so that it can handle multiple languages and cultural conventions *without redesign*. Internationalization takes place during design and development.

Software Localization Project Components



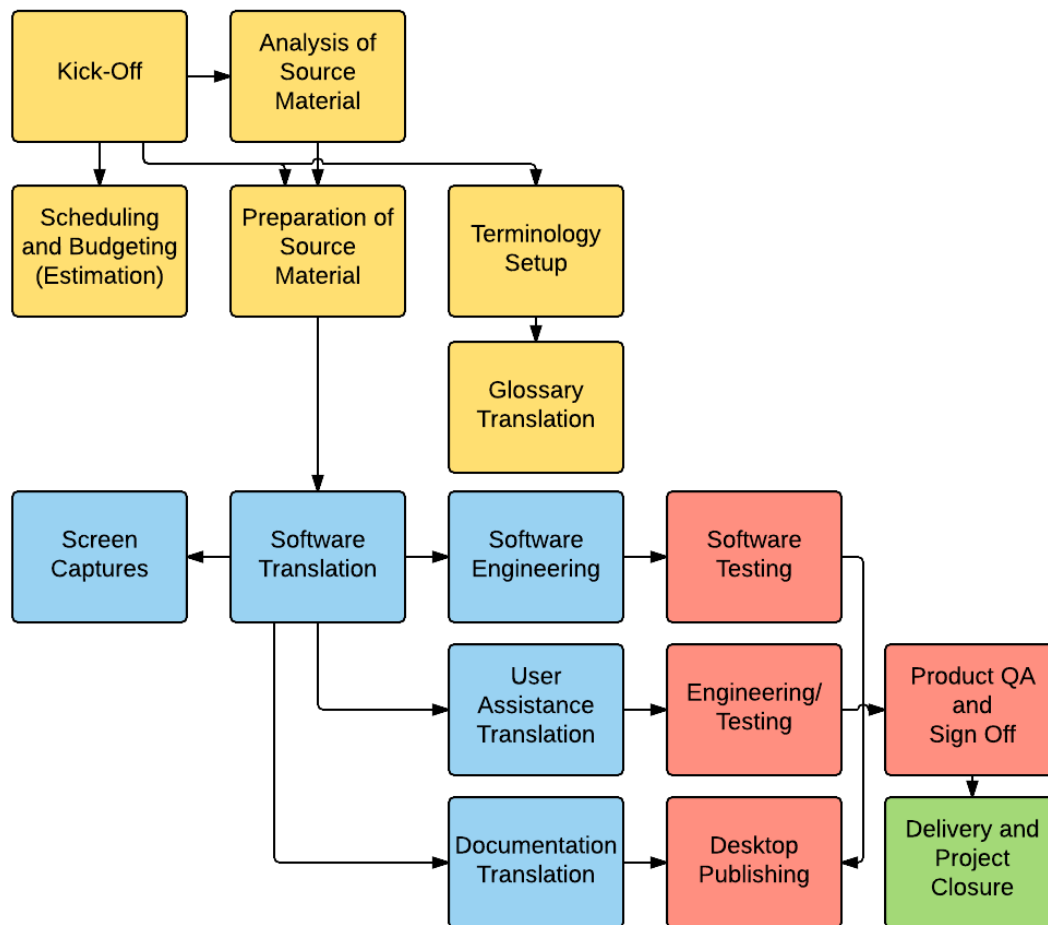
Many software products contain these components, and all of these components must be dealt with when it's time for localization. These include supportive materials in addition to the software application itself, such as user assistance (previously known as online help) and documentation of the product. Even though these components comprise one software product, they use different translation workflows according to file type.

Software Applications, for example, built with software strings for both user interface and platform will use translation management technology to support file types such as .resx, .rc, .js, .json, and more.

User Assistance systems will use a similar translation management technology that supports file types such as Flare, HTML or other UA file types.

Documentation also uses translation management technology that supports file types such as Microsoft Word, Adobe FrameMaker, XML or other types. After translation the files go to another workflow called Desktop Publishing (DTP), as the file types involve page layout applications and the translated content often needs major reformatting due to expansion or special attention due to character requirements. Documentation may include marketing collateral that are highly stylized art-filled pieces created with design applications such as InDesign and Illustrator.

Typical Software Localization Workflow



A typical software localization workflow includes these four steps, including preparation, translation, testing and delivery:

Step 1: **Project Kick-Off** provides analysis of source files, scheduling and budget (time and cost estimates provided by the translation vendor), preparation of source materials for translation and terminology setup in which a glossary will be created and translated first.

Step 2: **Software translation** initiates a series of workflows including software engineering (translation of software strings), **UA translation** (translation of user assistance content) and **document translation** (translation of documentation supporting the software product). Screen captures are created then translated or used as a guide for translation.

Step 3: **Testing** occurs after translated content is reintegrated into all files to identify errors. **Desktop Publishing (DTP)** tasks include reformatting tasks as mentioned in the previous section. **Product QA** is a comprehensive inspection to ensure components of the entire project perform as expected in

the final translated product. After quality assurance tasks are finished, the project is ready for sign off. An additional review round may be needed, involving in-country review (ICR) with subsequent revisions to be implemented.

Step 4: After final sign off the **Project is delivered** to the production team and the software product is launched. A good practice is to perform a post-mortem to identify future improvements, before **Project Closure**.

Success in Releasing Global Products

Software development projects may vary according to types of files or components used, but all are susceptible to errors if internationalization principles are not kept in mind while developing software.

Developers may be continually faced with tight deadlines to deliver functional software but with the use of guidelines and incorporating best practices, developers can facilitate localization processes that will ultimately help achieve success with global product releases.

Contact Us to Get Started

For more information and to begin localization and testing processes for your project, contact [Pearson Globalization Services](#).

Internationalization of the User Interface (UI)

Things to consider when developing a software user interface:

- Allow for text expansion by adding extra space to controls or buttons, but be aware of possible text contraction as well, *see Example 1*.
- Ensure all aspects of the UI fits on all screen resolutions as appropriate per platform, browser display vs. mobile display.



Example 1: Text expansion and contraction compared to English

Regional Standards

Most development environments support the *locale* model, that is, a collection of standard settings, rules and data specific to a language and geographical region. It's a good idea to be aware of differences such as:

- Sorting rules (English sorts A through Z, but many languages sort differently)
 - Traditional Chinese sorting is based on the number of strokes in the character.
 - Japanese sorting is based on phonetics.
 - In Swedish the letter Ä sorts after Z, not near the letter A.
- Measurement formats
 - Inches in the US, vs. centimeters in Europe and many other places across the globe.
- Number formats
 - Decimal separators and space separators, *see Example 2*.

- Time/date formats, *see Example 2*.
- Calendar, currency, address, and phone number formats.

English (US)	12:00 PM	1,234,567.89	12/31/2017
Spanish (Latin America)	12:00 p.m.	1,234,567.89	31/12/2017
Portuguese (Brazil)	14:00	1.234.567,89	31/12/2017
French (France)	14:00	1 234 567,89	31/12/2017
Traditional Chinese (Taiwan)	上午12:00 (AM) 下午12:00 (PM)	1,234,567.89	2017年12月31日

Example 2: Global formats for time, numbers and dates

Checklist for International Character Support

Ensure the product supports international sorting standards.

- ☐ Ensure the product is multibyte enabled, for example by using Unicode.
- ☐ Verify that users can successfully cut/paste text that contains accented (or double-byte/multibyte) characters to other applications.
- ☐ Verify that the application can open and save files with accented characters.
- ☐ Verify application can use bidirectional text and can display the appropriate mirror-flipped orientation of all elements as needed, *see Example 7*.
- ☐ Ensure the product supports international code sets, and supports input, output and display of accented characters.

Checklist for Translatable Files

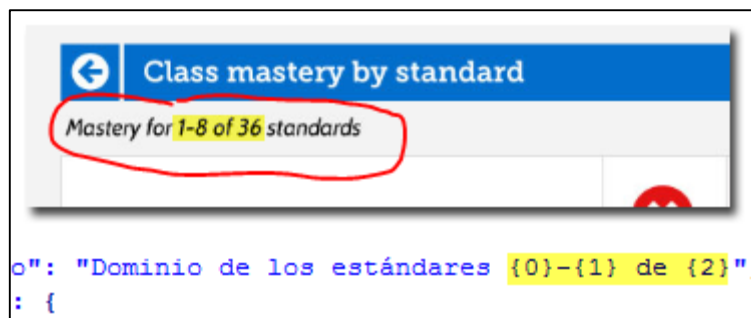
- ☐ Keep track of all resource files (images, icons) to be localized.
- ☐ Use variable names that are not “real” words to avoid translation of non-translatable strings. Make it obvious that the variable is not a real word.
- ☐ Exclude all non-translatable text strings from the resource files, unless they include comment text for translators.
- ☐ Use comments in code to provide description and context for translators, *see Example 3*.

```
<data name="TIMESAREDIS_68072" xml:space="preserve">
  <value>* Times are displayed in {0}</value>
  <comment>user's time zone</comment>
</data>
```

Example 3: Comments in code helps explain context

Checklist for UI Design

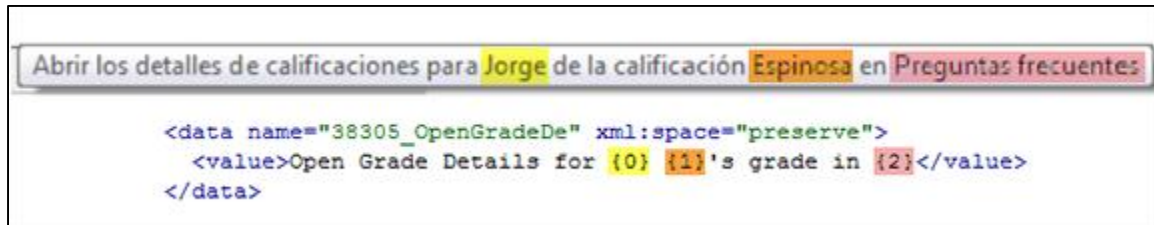
- ☐ Allow for text expansion; make buffers and UI large enough to accommodate translated text. French and German, for example, are usually up to 30% longer than the English originals.
- ☐ Be aware of text contraction as well in Asian languages. Watch for unexpected gaps from shorter line lengths after translation.
- ☐ Check the layout; both static and fluid layouts may have issues with expansion:
 - Static layouts may cut off text in long line lengths after translation
 - Fluid layouts can allow expansion, but long line lengths after translation can push elements into adjoining areas
- ☐ Make sure shortcut key combinations are accessible from international keyboards, for example, use Function keys instead of letters in shortcut key combinations, such as Ctrl+F3.
- ☐ Delete any obsolete text or strings from the resource files.
- ☐ Ensure custom, third-party controls in dialog boxes are localizable. This will depend on Pearson's agreement with the third-party vendors. In some cases, vendors will allow Pearson to translate the interface. In other cases, the UI may remain in English and any languages the vendor translates themselves.
- ☐ Be cautious about using icons or images showing people, hand gestures, holiday symbols, or body language. Use generic images that do not have text to be localized or adjusted for target locales.
- ☐ Limit the use of text or letters in images and icons.
- ☐ Avoid concatenated string issues (that is, combining strings at run-time to form new strings) by providing *full sentences* and providing reference material to translators such as screen captures, definitions, and comments in the code. Composite messages are often hard to translate so you'll need to provide references, *see Example 4*. Be open to changing wording as needed if suggested by translators.



Example 4: A screenshot helps translator understand context for {0}-{1} of {2}

Potential Issues if Concatenation is Not Handled Properly

1. Here's an example of a concatenated string issue that was not analyzed or fixed before release. The string variables are {0}, {1} and {2} which works in English, but because grammar and word order are different in other languages, the resulting translated rollover is incorrect, see *Example 5*.



Example 5: Rollover shows internationalization error due to concatenation

2. Here's another concatenated string issue because of two separate segments with a variable forming one sentence. This creates issues in Korean, so the translator left the first segment empty due to word order difference. This may result in an unexpected gap in the translated content, see *Example 6*.

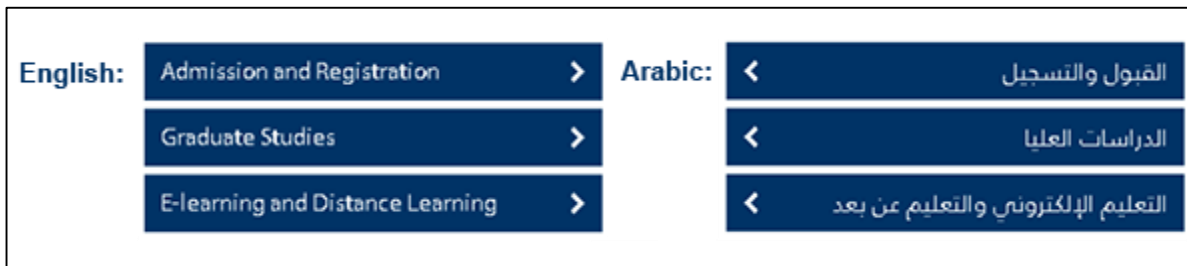
<value>Are the files in {0}</value>	(Intentionally left empty)
<value>linked to one another?</value>	안의 파일이 서로 링크되어 있습니까?

Example 6: Korean grammar issues require leaving a segment empty

Checklist for Locale Support

- ☐ Build in support for DBCS (Double-Byte Character Sets) for Asian languages, such as Japanese, Chinese, and Korean, to enable users to enter, store, process, retrieve, distribute, display, and print DBCS languages.
- ☐ Build in support for IME (Input Method Editor) handling for Arabic, Asian and Indic languages that are difficult to type using conventional keyboards. An IME is a set of tools that allow the user to type in their native language. Examples are virtual keyboards or built-in Pinyin keyboards (phonetic character entry).
- ☐ Build in support for bidirectional language use, including both language and layout mirroring as required, see *Example 7*.
- ☐ Ensure the product supports different calendars, date formats, and time formats.
- ☐ Ensure the product supports different number formats, currencies, weights, and measurement systems.

- ❑ Ensure the product supports international paper sizes, envelope sizes, and address formats.
- ❑ Use language support from the operating system running the software product.



Example 7: Bidirectional language support is needed for Arabic characters and reversed layout. This example shows the difference between English and Arabic alignment

Language Use in Software Projects

Software deployment teams may use certain practices in the handling of translated content such as redirection of files. For example, a common practice is to serve multiple Latin American Spanish language locales with one generic ES-LA language file. *Please check with the translation team first.*

Project Glossary

Part of the translation project setup is to create a glossary at project start. The simplest version could be a spreadsheet with English source words in the first column, definitions of the terms in the next, then translations in subsequent columns for each target language. Include previously translated terms, preferred translated terms, abbreviations and words that stay in English. Be aware of other Pearson platforms terminology and consider aligning with them to save on costs and improve consistency.

Pearson Translation Request Form

Each project should have a translation request form included along with software files and resource files as part of translation preparation. List file types, file names, and any details or requirements needing to be communicated pre-translation.

Continuous Translation

On-going translation processes are being used in a few Pearson divisions that are more efficient than previous methods. This process allows direct integration of Jira into translation vendor workflows.

Contact us for more information

For more information on all these topics, contact [Pearson Globalization Services](#). You can get a translation request form, a project glossary template and find out how to engage the Translation/Localization team.

Testing and Quality Assurance (QA)

A common practice is to skip testing and QA when a software project is released, due to lack of time or budget. This can lead to costly errors and a perceived failure of the product in its intended market. The best way to avoid these issues is to use the Localization Testing team both *during* development and *before* release. Here are steps to building a solid quality assurance process for your project.

1. **QA Access:** Make sure the QA team has access to the platform or production server, are able to switch languages easily and can view the appropriate content. (For example, do they need both Instructor and Student logins to view separate content.)
2. **QA List:** Identify tasks to be performed and define who does what.
 - a. List tasks to be performed by developers, such as:
 - Demonstrating to the testing team how to use the platform
 - Define areas to be tested and what not to test
 - b. List tasks to be performed by linguists, such as:
 - Creation of translated email addresses and instructor/student names for testing
 - Testing of platform
3. **Quality Assurance Goals:** Agree upon and document goals for testing (Functional, linguistic, and so on.)
4. **Levels of Testing:** Agree to what degree of testing is required
 - a. If the system is soon to be retired, then do only a light level; if the system is new there should be comprehensive testing.
 - b. *Pseudolocalization* can be used to test before translation is performed as a low-cost method to reveal issues.
5. **Defect Tracking:** Define best methods to record issues, communicate to teams and resolve issues. Our primary vendor, SDL, uses an online bug tracking system that supports comments and screenshots and automatically routes to appropriate teams.
6. **Resources and Scheduling:** Determine scope, assign teams, map out schedules.
7. **Deliverables:** Agree to terms of delivery items, who enters changes, reintegration, retesting then completion.

Contact us for QA and Testing

These steps are listed in Pearson Globalization Service's *Localization QA Plan template*. Contact [Pearson Globalization Services](#) for a copy of the Localization QA Plan template and to engage the Localization Testing team.

Appendix A — Language Codes and Regions

The following table lists language codes and the countries where they are used. These codes are used when defining language requirements for each project instead of the language name. This list is not exhaustive but covers many of the languages encountered by Pearson.

<i>Language Codes and Regions for Localization</i>			
ar-EG	Arabic - Egypt	id-ID	Indonesian - Indonesia
ar-SA	Arabic - Saudi Arabia	it-IT	Italian - Italy
ar-AE	Arabic - United Arab Emirates	ja-JP	Japanese - Japan
zh-CN	Chinese - China, PRC	ko-KR	Korean - Korea
zh-TW	Chinese - Taiwan	ms-MY	Malay - Malaysia
da-DK	Danish - Denmark	nn-NO	Norwegian (Nynorsk) - Norway
nl-NL	Dutch - Netherlands	pl-PL	Polish - Poland
en-AU	English - Australia	pt-BR	Portuguese - Brazil
en-ZA	English - South Africa	pt-PT	Portuguese - Portugal
en-GB	English - Great Britain	ru-RU	Russian - Russia
en-US	English - United States	es-AR	Spanish - Argentina
fa-IR	Farsi - Iran	es-CL	Spanish - Chile
TL-fil	Filipino - Tagalog	es-CO	Spanish - Colombia
fi-FI	Finnish - Finland	es-MX	Spanish - Mexico
fr-CA	French - Canada	es-PR	Spanish - Puerto Rico
fr-FR	French - France	es-ES	Spanish - Spain
de-DE	German - Germany	sv-SE	Swedish - Sweden
el-GR	Greek - Greece	th-TH	Thai - Thailand
hi-IN	Hindi - India	tr-TR	Turkish - Turkey
hu-HU	Hungarian - Hungary	vi	Vietnamese

Appendix B — Localization Definitions

ANSI

Abbreviation for American National Standard Organization; also used as the short form for the 8-bit ANSI code, which is a character set of 256 characters. The first 128 are the same for all countries; the higher 128 differ. Unicode is preferred for use in translation projects due to these limitations. [See Unicode]

ASCII

The American Standard Code for Information Interchange, a character-encoding scheme originally based on the English alphabet that encodes 128 specified characters.

Bidirectional (bidi)

A mixture of characters that are read from left-to-right and characters that are read from right-to-left. Most Arabic and Hebrew characters, for example, are read from right-to-left, but numbers and quoted Western terms within Arabic or Hebrew text are read from left-to-right.

Language Codes

Short alphabetic codes developed to represent languages and countries, used in the translation industry as unique identifiers for each language. Some codes are chosen based on the native names of the countries. For example, Germany is assigned the alpha-2 code DE, based on its native name "Deutschland". [See Appendix A]

Double-byte, double-byte character set (DBCS)

A character set that is defined with two bytes (16 bits) instead of one byte (8 bits). Thus, you can define two power 16 = 65,536 characters, instead of the single byte and 256 characters stored in ANSI. Examples of double-byte languages include Japanese, Korean, and Chinese. [Similar to Multi-byte]

Exact match

Used to describe matching with a translation memory. Also known as 100% match and repetitions. This is a segment stored in the translation memory, identical to the source segment that is being translated.

Fuzzy match

Used to describe matching within a translation memory. A segment that is similar (but not identical) to the sentence or phrase that translator is currently translating.

Globalization, globalisation (UK), G11n

Globalization is the result of combining Internationalization plus Localization.
[See Internationalization and Localization]

Glossary

In the context of localization, a glossary is a list of source language terms paired with a list of corresponding terms in the target language.

Graphical User Interface (GUI)

The visible part of an application. A GUI is mostly organized in forms having menus, buttons, frames, dialog boxes and much more.

In-Country Review (ICR)

Pearson content experts in each target country review translated materials for style, tone and branding accuracy.

Internationalization, internationalisation (UK), i18n

Internationalization is the process of generalizing a product so that it can handle multiple languages and cultural conventions without the need for redesign.

Linguistic Signoff (LSO)

Review of content after translations are reintegrated back into native format file types. LSO is completed only for formattable resources. Other content types have review through linguistic testing/bug tracking.

Localization, localisation (UK), L10n, localize, localise (UK)

The process of adapting products and services (web sites, manuals and software) in accordance to linguistic, cultural, technical and other locale-specific requirements of the target market. It means much more than plain translation. L10n covers date and time formats, numeric formats and more.

Machine translation (MT)

Automated technology that translates text from one human language to another, using terminology glossaries and advanced grammatical, syntactic, and semantic analysis techniques. However the automatic machine translation systems available today still are not able to produce high-quality translations unaided; their output must be edited by a human to correct errors and improve the quality of translation.

Multi-byte character set (MBCS)

An alternative to Unicode for supporting character sets like Japanese and Chinese that cannot be represented in a single byte. A multi-byte character set may consist of both one-byte and two-byte characters. Thus a multi-byte character string may contain a mixture of single-byte and double-byte characters.

New text

Used to describe matching with a translation memory. Text where the source segments being translated do not correspond to any of the target segments.

Resource files

Source files that contain information to be compiled into the program. They contain the parts of the application that is seen by the user.

Right-to-left languages

Languages such as Hebrew, Arabic, Urdu, and Farsi are written primarily right to left.

Segment

The basic unit of source text as identified by a translation tool that can be aligned with a corresponding translation from the translation memory. A segment is commonly defined as the content from one paragraph break to the next, usually a sentence, but a segment can be a header, items in a list, cells in a table, a paragraph, and so on.

Terminology list

Created as a reference for the linguists (translators) with the English source word and the target language equivalent for the project.

Terminology management

Primarily concerned with manipulating terminological resource for specific purposes, such as establishing repositories of terminology resources for publishing dictionaries, maintaining terminology databases, and so on.

Text expansion

The increase in the total number of characters that often occurs during translation.

Translate, translation

Translation is the process of changing text from one language to another.

Translation Management System (TMS)

A Translation Management System (TMS) is a system that manages project information, process workflow, and language assets required for large-scale translation activity.

Translation Memory (TM)

Translation Memory is a database which stores combination of source text and translations. Using translation memory makes it easy to reuse existing translation efforts in updates or other related translation projects. Proper use of TM can lead to time and cost savings.

Unicode

A replacement for character sets like ANSI. Character sets like ANSI work with code pages to support multiple languages therefore cannot display Russian and Chinese text on the same form, for example. Unicode can do this because it has space for more than 256 chars like ANSI.

UTF-8, UTF-16, UTF-32

These are variable-length character encodings for Unicode. However, in UTF-8 a character may occupy a minimum of 8 bits, while in UTF-16 character length starts with 16 bits, and UTF-32 is a 32-bit Unicode transformation format. Depending on the encoding form you choose (UTF-8, UTF-16, or UTF-32), each character will then be represented either as a sequence of one to four 8-bit bytes, one or two 16-bit code units, or a single 32-bit code unit.

Sources for this guide:
The Localization Guide, Bert Esselink
Certification Materials, The Localization Institute

Contact Us for More Information

Marina Gracen-Farrell, CLP, CLPM
marina.gracen-farrell@pearson.com

Pearson Globalization Services
GlobalizationRequest@Pearson.com

Join us on Neo

Join the **Pearson Globalization Services** group on Neo for articles, localization guides, and to find out more.

<https://neo.pearson.com/groups/pearson-globalization>