

# **A Practical Guide to Localization**

The LANGUAGE INTERNATIONAL WORLD DIRECTORY is a series of international listings on subjects pertaining to language related practice such as language policy and planning, training, translation, modern tools for teaching, lexicography, terminology, etc.

The LANGUAGE INTERNATIONAL WORLD DIRECTORY is published under the auspices of *Language International: The magazine for language professionals*.

#### THE LANGUAGE INTERNATIONAL WORLD DIRECTORY

1. *Language International World Directory of Sociolinguistic and Language Planning Organizations* compiled by Francesc Domínguez and Núria López, 1995.
2. *Language International World Directory of Translation and Interpreting Schools* compiled by Brian Harris, 1997.
3. *A Practical Guide to Software Localization* by Bert Esselink, 1998.
4. *A Practical Guide to Localization* by Bert Esselink, 2000.

*Volume 4*

Bert Esselink

*A Practical Guide to Localization*

# A Practical Guide to Localization

---

Bert Esselink

Development Editor: Arjen-Sjoerd de Vries

Copy Editor: Shiera O'Brien

JOHN BENJAMINS PUBLISHING COMPANY  
AMSTERDAM/PHILADELPHIA



The paper used in this publication meets the minimum requirements of the American National Standard for Information Sciences – Permanence of Paper for Printed Library Materials, ANSI Z39.48-1984.

#### Library of Congress Cataloging-in-Publication Data

Esselink, Bert

A Practical Guide to Localization / Bert Esselink.

Rev. ed. of: A practical guide to software localization

p. cm. (Language International World Directory, ISSN 1383-7591 ; v. 4)

Includes bibliographical references and index.

1. Software localization 2. Software documentation. I. Esselink, Bert. A practical guide to software localization I. Title II. Series.

QA76.76.D63E87 2000

005.1 -- dc 21

00-058639

ISBN 978 90 272 1955 8 (EUR) / 978 1 58811 005 3 (US) (Hb ; alk. paper)

ISBN 978 90 272 1956 5 (EUR) / 978 1 58811 006 0 (US) (Pb ; alk. paper)

ISBN 978 90 272 9818 8 (Eb)

© 2000 – John Benjamins B.V.

No part of this book may be reproduced in any form, by print, photoprint, microfilm, or any other means, without written permission from the publisher.

John Benjamins Publishing Co. · P.O. Box 36224 · 1020 ME Amsterdam · The Netherlands  
John Benjamins North America · P.O. Box 27519 · Philadelphia PA 19118-0519 · USA

# Preface

The preface of the first edition of this book started with “Writing about software localization is like fighting against time.” In the two years that have passed since the first edition was published, the localization industry has changed quite dramatically. The web has taken over the world and new, web-based localization providers are challenging the larger, established localization vendors by offering advanced translation workflow automation solutions. Translation technology has evolved quickly, with an even greater number of translation tools available than ever before. Global or local web sites have become a necessity rather than a luxury, and more than half of all web pages are now in languages other than English.

Many of these changes are reflected in the second edition of *A Practical Guide to Software Localization*. For this second edition, I have elected to omit the word “Software” from the title, because the book covers a wide variety of topics useful to people localizing and translating material in other areas, such as web sites or “traditional” documentation.

In this new and totally revised edition, a lot of information has been added on translating and engineering HTML and XML documents, multilingual web sites, and HTML-based online help systems. Other important changes include the addition of new chapters on internationalization, software quality assurance, and desktop publishing.

Because information on localization becomes outdated very quickly, some sections from the book have been moved to the web. At [www.locguide.com](http://www.locguide.com) you will find many links to useful online resources related to localization and translation. The web site also contains updates to the book, exercises, and some sample files.

A big thank you to my colleagues at International Software Products, ALPNET, and Lionbridge Technologies for everything they have taught me over the past years. Special thanks to Amancio Quant for designing the new layout, Alex Eamis and Carmen Andrés Lange for reviewing some of the chapters, and Arjen-Sjoerd de Vries, Bob Clark, and Sue Ellen Wright for volunteering to review the book in its entirety. Shiera, thanks for doing such a great job copy-editing the book.

Last but never least, thank you Caroline for your continuing support and patience.

Bert Esselink, Haarlem 2000



# Table of Contents

- CHAPTER 1:**
  - INTRODUCTION..... 1**
    - 1. Introduction..... 1
    - 2. Definitions of Terms ..... 2
    - 3. Localization Industry..... 5
    - 4. Localization Projects..... 10
    - 5. Further Reading..... 24
  
- CHAPTER 2:**
  - INTERNATIONALIZATION ..... 25**
    - 1. Introduction..... 25
    - 2. Language ..... 27
    - 3. Software..... 31
    - 4. Web Sites and HTML ..... 36
    - 5. Documentation..... 43
    - 6. Creating Localization Kits ..... 48
    - 7. Further Reading..... 54
  
- CHAPTER 3:**
  - SOFTWARE TRANSLATION ..... 57**
    - 1. Introduction..... 57
    - 2. Translation Guidelines ..... 65
    - 3. Windows..... 76
    - 4. Mac OS ..... 85
    - 5. Further Reading..... 95
  
- CHAPTER 4:**
  - SOFTWARE ENGINEERING ..... 97**
    - 1. Introduction..... 97
    - 2. Evaluation and Preparation ..... 99
    - 3. Compiling and Engineering..... 106
    - 4. Windows..... 113
    - 5. Mac OS ..... 127
    - 6. Other Environments..... 137
    - 7. Further Reading..... 143
  
- CHAPTER 5:**
  - SOFTWARE QUALITY ASSURANCE..... 145**
    - 1. Introduction..... 145
    - 2. Levels of Testing ..... 147

- 3. Preparation..... 155
- 4. Test Management ..... 160
- 5. Testing Tools ..... 163
- 6. Further Reading ..... 164

**CHAPTER 6:**

**ONLINE HELP TRANSLATION..... 165**

- 1. Introduction ..... 165
- 2. Windows ..... 172
- 3. Ma cOS..... 195
- 4. HTML and XML..... 205
- 5. Readme Files..... 223
- 6. Further Reading ..... 224

**CHAPTER 7:**

**ONLINE HELP ENGINEERING AND TESTING ..... 225**

- 1. Introduction ..... 225
- 2. Evaluation and Preparation..... 227
- 3. Windows ..... 233
- 4. Ma cOS..... 252
- 5. HTML and XML..... 263
- 6. Testing Localized Online Help ..... 278
- 7. Further Reading ..... 287

**CHAPTER 8:**

**DOCUMENTATION TRANSLATION..... 289**

- 1. Introduction ..... 289
- 2. Translating Manuals..... 293
- 3. Translating Collateral Material ..... 309
- 4. Reviewing Documentation ..... 315
- 5. Further Reading ..... 321

**CHAPTER 9:**

**DESKTOP PUBLISHING..... 323**

- 1. Introduction ..... 323
- 2. Evaluation and Preparation..... 325
- 3. Formatting and Page Layout ..... 334
- 4. Documentation Output ..... 340
- 5. Further Reading ..... 346

**CHAPTER 10:**

**GRAPHICS LOCALIZATION..... 347**

- 1. Introduction ..... 347
- 2. Creating Screen Captures ..... 349
- 3. Editing Graphics ..... 353



<b>CHAPTER 11:</b>	
<b>TRANSLATION TECHNOLOGY.....</b>	<b>359</b>
1. Introduction.....	359
2. Translation Memory Tools.....	362
3. Terminology Tools.....	379
4. Software Localization Tools.....	383
5. Machine Translation Tools .....	394
6. Word Counting Tools .....	396
<b>CHAPTER 12:</b>	
<b>TERMINOLOGY .....</b>	<b>397</b>
1. Introduction.....	397
2. Terminology Setup.....	400
3. Terminology Reference Materials.....	401
4. Terminology Standards.....	407
5. Creating User Interface Glossaries .....	409
6. Further Reading.....	411
<b>CHAPTER 13:</b>	
<b>PROJECT EVALUATIONS.....</b>	<b>413</b>
1. Introduction.....	413
2. Activities .....	414
3. Word Counts .....	418
<b>CHAPTER 14:</b>	
<b>PROJECT MANAGEMENT.....</b>	<b>427</b>
1. Introduction.....	427
2. Tasks and Responsibilities.....	428
3. Organizations and Training.....	461
4. Tools and Applications.....	463
5. Further Reading.....	466
<b>APPENDIX A:</b>	
<b>GLOSSARY OF TERMS.....</b>	<b>467</b>
<b>APPENDIX B:</b>	
<b>LOCALIZATION IN THE NEW MILLENNIUM.....</b>	<b>477</b>
1. Localization Models.....	477
2. Translation Technology .....	478
3. Localization Industry.....	480
4. Structured Translation.....	481
<b>INDEX.....</b>	<b>483</b>



# Chapter 1: Introduction

This introduction contains information on the basic principles of localization, including definitions for commonly used terms, an overview of the localization industry, and a description of a typical localization project, which serves as an outline for the rest of this book.

This chapter includes the following sections:

1. [Introduction](#)
2. [Definitions of Terms](#)
3. [Localization Industry](#)
4. [Localization Projects](#)
5. [Further Reading](#)

## 1 Introduction

Over the past decades, localization has progressed from being an added effort by some software publishers to a multi-billion dollar professional industry. Localization, web site globalization, language engineering, and software internationalization have become important issues for companies that want to market and sell their products in international markets. In many cases, localization has proven to be the key factor for international product acceptance and success.

As explained later in the Definitions section, many different definitions exist for the term “localization”. Generally speaking, localization is the translation and adaptation of a software or web product, which includes the software application itself and all related product documentation. The term “localization” is derived from the word “locale”, which traditionally means a small area or vicinity. Today, locale is mostly used in a technical context, where it represents a specific combination of language, region, and character encoding. For example, the French spoken in Canada is a different locale to the French spoken in France.

Before the Internet and World Wide Web transformed software development and localization, a typical localization project would encompass full translation and engineering of a software application, its online help files, a set of printed manuals, and reference and registration cards included in the product box. Many current localization projects still fit this description. However, because of new web-based publishing and distribution technologies, localization can now also include the translation and adaptation of web-based applications and database-driven web sites.

Traditionally, translation is only one of the activities in a project where material is transferred from one language to another. Other activities in traditional translation projects include terminology research, editing, proofreading, and page layout. In localization, many more activities have been added to this list. Examples of activities in localization which are not necessarily part of traditional translation include multilingual project management, software and online help engineering and testing, conversion of translated documentation to other formats, translation memory alignment and management, multilingual product support, and translation strategy consulting.

Most large, multi-language localization agencies focus on these additional activities, while outsourcing core translation activities to freelance translators. Only final language quality assurance is performed in-house by these vendors.

Another key difference between localization and translation is the fact that traditional translation is typically an activity performed after the source document has been finalized. Localization projects, on the other hand, often run in parallel with the development of the source product to enable simultaneous shipment of all language versions. For example, the translation of software strings may often start while the software product is still in beta phase.

The key reasons why software publishers localize their products are local market and legal requirements. In most countries, computer users prefer to work with software in their native language. In order to increase sales opportunities in target countries, software publishers have to localize their products, while local law often requires all imported hardware or devices to be accompanied by a user manual in the local language.

## 2 Definitions of Terms

Many different definitions and descriptions for *localization* are used in the software and translation industry. Below you will find a commonly used definition, as well as descriptions of some related terms, including: Internationalization, Localization, Globalization, Translation, and Language Engineering.

Explanations of many other commonly used terms can be found in Appendix A, Glossary of Terms.

### 2.1 Internationalization

The Localisation Industry Standards Association (LISA) defines internationalization as follows:

“Internationalization is the process of generalizing a product so that it can handle multiple languages and cultural conventions without the need for re-design. Internationalization takes place at the level of program design and document development.”

In general, a product is internationalized during the product development cycle, as a precursor to the localization of a product.

An important aspect of internationalization is the separation of text from the software source code. Translatable text, i.e. text which is visible to the user, should be moved to separate strings-only resource files. This will prevent translators from changing – or breaking – the program code, because the resource files only contain translatable components and no coding.

Central to internationalization is the ability to display the character sets and support local standards of a particular language and country. For example, before a software product can be translated into Japanese, it must support double-byte characters. If the application has been programmed to support only Western European languages, it must be *double-byte enabled* first, for example by using Unicode character support. Likewise, different foreign keyboard layouts, input methods, and hardware standards must be supported.

Internationalization is not limited to software. Online help, documentation, and web sites, in particular, also need to be internationalized. For technical writers this process is often called “writing for translation” or “writing for a global audience”. In the case of web sites, internationalization, translation, and adapting the content to specific target markets is usually referred to as “web site globalization.”

Internationalization is often abbreviated to “i18n”, where “18” indicates the number of letters between the “i” and the “n”. For more information on internationalization, refer to Chapter 2, Internationalization.

## 2.2 Localization

The Localisation Industry Standards Association (LISA) defines localization as follows:

“Localization involves taking a product and making it linguistically and culturally appropriate to the target locale (country/region and language) where it will be used and sold.”

Note that some publishers consider localization as an integral part of the development process of a product. In some cases, special country-specific releases of software products are called *localizations*. In this book, we will refer to all localization-related activities taking place during development of the original product as *internationalization*.

Localization projects usually include the following activities:

- Project management
- Translation and engineering of software
- Translation, engineering, and testing of online help or web content
- Translation and desktop publishing (DTP) of documentation
- Translation and assembling of multimedia or computer-based training components
- Functionality testing of localized software or web applications

Approximately 80% of software products are localized from English into other languages because the majority of software and web applications are being developed in the United States. In addition, software manufacturers in other countries often develop their products in English, or have them localized into English first and use this version as a basis for further localization.

A well-localized product enables users to interact with a software application in their native language. They should be able to read all interface components such as error messages or screen tips in their native language, and enter information with all accented characters using the local keyboard layout. “L10n” is often used as an abbreviation for localization.

## 2.3 Globalization

The Localisation Industry Standards Association (LISA) defines globalization as follows: “Globalization addresses the business issues associated with taking a product global. In the globalization of high-tech products this involves integrating localization throughout a company, after proper internationalization and product design, as well as marketing, sales, and support in the world market.”

Globalization is a term used in many different ways. For example, there is the top, geopolitical level that deals with globalization of business as an economic evolution. Secondly, there is the globalization of an enterprise that establishes an international presence with local branch or distribution offices. Thirdly, there is the process of creating local or localized versions of web sites, which we will refer to as “web site globalization”. Web site globalization refers to enabling a web site to deal with non-English speaking visitors, i.e. internationalizing the site’s back-end software, designing a multi-lingual architecture, and localizing the site’s static or dynamic content.

In the context of this book, globalization covers both internationalization and localization. Publishers will “go global” when they start developing, translating, marketing, and distributing their products to foreign language markets. The concept of globalization (“g11n”) is typically used in a sales and marketing context, i.e. it is the process by which a company breaks free of the home markets to pursue business opportunities wherever its customers may be located.

## 2.4 Translation

Translation is the process of converting written text or spoken words to another language. It requires that the full meaning of the source material be accurately rendered into the target language, with special attention paid to cultural nuance and style.

The difference between translation and localization can be defined as follows:

“Translation is only one of the activities in localization; in addition to translation, a localization project includes many other tasks such as project management, software engineering, testing, and desktop publishing.”

In localization there is stronger emphasis placed on translation tools and technology compared to the traditional translation industry.

## 2.5 Language Engineering

The following definition for Language Engineering was extracted from The Euromap Report published on behalf of the EUROMAP Consortium in 1998:

“Language engineering is the application of knowledge of written and spoken language to the development of information, transaction and communication systems, so that they can recognise, understand, interpret, and generate human language. Language technologies include, for example, automatic or computer assisted translation, speech recognition and synthesis, speaker verification, semantic searches and information retrieval, text mining and fact extraction.”

Language engineering and computer aided translation (CAT) are discussed briefly in this book. Most language engineering applications are not yet utilized on a large scale in the localization industry, even though this is changing rapidly. Many language engineering techniques such as sentence parsing, terminology extraction, or text mining are now integrated in CAT tools used in localization.

A company that is very active in the language engineering field is Lernout & Hauspie ([www.lhsl.com](http://www.lhsl.com)). Another good resource for more information on language engineering is the Human Language Technologies web site at [www.linglink.lu](http://www.linglink.lu).

## 3 Localization Industry

The localization industry is relatively young. Until the early 1980s, U.S. based software publishers did not appreciate the need for internationalized and localized products. This has changed dramatically since the early 1990s. The growth of the Internet, in particular, has made it much easier for software publishers to market and distribute their products in other countries.

Business reasons aside, many companies are translating and localizing their products for legal reasons. In many countries, such as some Baltic countries, importing or even using products which are not in the country's native language is not permitted.

### 3.1 History

In the early 1980s, most software publishers either started in-house translation departments, or outsourced translation work to freelance translators or in-country product distributors.

The increasing size and complexity of localization projects soon forced companies to an outsourcing model. Most software publishers simply did not have the time, knowledge or resources to manage multilingual translation or localization projects.

In the mid 1980s, the first multi-language vendors (MLVs) were formed. New companies such as INK (now Lionbridge) or IDOC (now Bowne) specialized in the management and translation of technical documentation and software. Existing companies with other core competencies, such as Berlitz, started translation divisions that could handle multilingual translation and localization projects.

At the same time, many software publishers were still experiencing bottlenecks in their multilingual product release cycles. For example, in their engineering and testing departments, they were suddenly faced with the task of testing multiple language versions instead of just one English version. As a result, an extended outsourcing model took off in the beginning of the 1990s. In addition to translation services, MLVs then began offering engineering, testing, desktop publishing, printing, and support services.

This development kickstarted the transition from translation into localization. In 1990, the Localisation Industry Standards Association (LISA) was founded, in an attempt to bring together the IT industry and localization service providers.

In the second half of the 1990s, the localization industry began to consolidate. Since the mid 1990s, the 30 largest translation and localization service providers have been reduced to a dozen multi-language providers with a global presence. Examples of these MLVs are Lionbridge Technologies, ALPNET, and Berlitz GlobalNET.

### 3.1.1 SERVICE PROVIDERS

Today, most software publishers outsource translation and localization activities to localization service providers. In the 1980s, large in-house localization divisions were commonplace in software firms. These divisions were either closed or reduced to just a coordination or quality assurance department.

A localization manager or vendor manager is now often the key person at the publisher site, serving as the link between the development department and the localization service provider.

### 3.1.2 OUTSOURCING

Most large software or web publishers today are interested in working with multi-language vendors (MLV) which can take charge of a range of target languages and language-related activities. MLVs are specialized firms, very often with expertise in both language services and technology. The main benefit to this approach is that project management and technical activities are typically centralized in one location by the vendor. Benefits for publishers are that they can focus on their core competencies and have one point of contact for all languages as opposed to a different one per language.

Despite consolidation activity, many single-language vendors (SLVs) are still active in the localization industry. Most SLVs specialize in only one target language and may have limited technical capacity. Many of them work with MLVs, or may have signed strategic partnerships with MLVs that guarantee business volumes in exchange for an exclusive business relationship.

Another approach often preferred or combined with the use of MLVs is risk spreading, where publishers divide – sets of – languages or products between different localization service providers to spread the risk if projects go wrong and to avoid bottlenecks when many languages are involved. Because localization service providers want to be able to offer as many languages as possible, many of them have subsidiaries or partners (often SLVs) in countries where no wholly-owned offices exist.



### 3.1.3 CONSOLIDATION

In the second half of the 1990s, major mergers and acquisitions took place in the localization industry, which created about a dozen service providers with offices spread out across the world. Not only have many global localization companies acquired smaller ones; many partnerships were also formed to create worldwide service networks such as Lernout & Hauspie, ALPNET, Lionbridge, and Berlitz GlobalNET.

Examples of consolidations that took place in the late 1990s are the acquisitions of LMI by Berlitz, ITP by SDL, and INT'L.com by Lionbridge.

### 3.1.4 IRELAND

Even though it seems localization vendors are now moving activities to many locations across the globe, Ireland established itself as the leader in the localization industry during the 1990s. Over the past 10 to 20 years, the Industrial Development Authority (IDA) had the mandate as a semi-governmental body to move Ireland forward industrially by attracting foreign investment. Back in the 1980s, a high concentration of manufacturing type companies started in Ireland, including some hi-tech companies. The Irish government at the time was providing what they called turnkey factories, where a large multinational was offered a certain amount of government subsidy per employee, plus facilities and grants, and a corporate tax rate of 10%, as an incentive to invest in Ireland.

After some failed investments and the increased competition from manufacturing in cheap labour markets, the Irish government changed strategy to focus on Research and Development and the hi-tech, blue-chip companies, i.e. with a more long-term strategy in mind. Most large software and web companies now have a presence in Ireland, with the bulk of their localization being managed from there, including Microsoft, Oracle, Lotus Development, Visio International, Sun Microsystems, Siebel, and FileNET.

The key benefits they offered these companies included:

- Government grants – A certain amount of money per employee, 10% corporate tax rate, exemption from VAT (Value-Added Tax). All products, including software, exported to Europe are exempt from VAT in Ireland.
- Competitive labour costs – With social costs at approximately 12-15% per employee, it is cheaper to employ in Ireland than in many of the European Union countries. Compared to the United States, development costs are still lower in Ireland.
- A young, well-educated, and motivated work force – Approximately 50% of the population was under 25 at the beginning of the 1990s.

The Irish government has invested a great deal of subsidy in education. There now is a strong push to offer additional computer courses to cope with the growing demand for IT and localization staff. This, combined with the fact that Ireland is an English speaking nation on the edge of Europe that serves as a gateway to Europe and the Euro zone, made many U.S. based companies decide to base their European headquarters or distribution centers in Dublin.

## 3.2 Market Developments

Historically, the largest markets for localized products have been France, Germany, and Japan. Medium-sized markets include Brazil, Italy, Spain, Sweden, Norway, and the Netherlands. Software publishers typically localize their products into FIGS (French, Italian, German, Spanish) and Japanese first. Swedish, Norwegian, Danish, Dutch, or Brazilian Portuguese often follow as second tier languages.

Because the localization industry is growing, on average by 30% annually, the need for localization specialists is more obvious than ever. Translators need a greater knowledge of computer technology, and engineers better language skills. Many translation schools and universities recognize this market potential and are now offering localization courses.

Inevitably, the localization industry will mature over the coming years. Competition in the localization industry is very strong, and service providers will have to distinguish themselves by offering value-added services such as internationalization consulting, functional software testing, multilingual product support, and structured document authoring and management. At the same time, many software publishers are moving towards “total outsourcing” models, where all globalization and localization-related activities are outsourced to large, multi-language localization vendors.

## 3.3 Organizations

In 1990, the Localisation Industry Standards Association (LISA) was founded in Switzerland. LISA defines its mission as “promoting the localization and internationalization industry and providing a mechanism and services to enable companies to exchange and share information on the development of processes, tools, technologies and business models connected with localization, internationalization and related topics”.

LISA organizes forums and regular workshops, where members can share information on the latest developments in the industry and attend training workshops. For more information, visit the LISA web site at [www.lisa.org](http://www.lisa.org).

In Ireland, two organizations were founded in the nineties to establish contacts between software publishers, localization service providers, and universities in and around the Dublin area. The Localisation Research Centre (LRC) was established at the University of Limerick in April 1998 as the result of a merger between the Centre for Language Engineering and the Localisation Resources Centre. The Software Localisation Interest Group (SLIG) is a special interest group for all parties involved in localization, and focuses specifically on the business interests of the localisation industry in Ireland and was founded in February 1984. For more information on LRC and SLIG visit their web sites at [lrc.csis.ul.ie](http://lrc.csis.ul.ie) and [www.slig.ie](http://www.slig.ie) respectively.

## 3.4 Technology

The late 1990s have also brought important changes in international software and web development. In many current software applications, users are able to produce multilingual content without any special international software support or add-ons.

Software engineers also seem to be more aware of internationalization issues and are learning how to integrate international support in their products.

One important technology development that has affected the localization industry to a large extent is the Unicode standard. Unicode offers a solution for character set issues, i.e. it offers support for all scripts that are being used around the world today. The Unicode standard uses two bytes (16 bits) for all characters instead of just one byte for standard European characters and has been implemented throughout the Microsoft Windows platforms, Apple Mac OS, and other computing platforms. For more information on Unicode, visit the Unicode Consortium web site at [www.unicode.org](http://www.unicode.org).

Another important change in international software development has been the introduction of a “single world-wide binary”, i.e. development of one version of a program that supports all languages. This single binary is often combined with “resource-only .dll files”, where all user interface text elements, such as dialog box options, menus and error messages, are centralized. All program code is separated from the resources, which means that applications can be run in another language by replacing the resource-only .dll with a localized version. A good example of this technology can be found in Microsoft Windows 2000, and Microsoft Office 2000, where users can run the applications using the user interface in their preferred language and preferred regional settings without impacting the functionality.

## 3.5 Training

Until quite recently, finding language graduates with computer or localization skills was virtually impossible. All translation and localization service providers had to develop internal training for new employees on localization-specific computer usage and translation tools.

Linguists working for a localization vendor are normally expected to have the following skills or qualifications:

- Native speakers of their target language
- Advanced knowledge of computer applications
- Experience using computer aided translation tools, e.g. translation memory
- Subject matter knowledge, e.g. expertise in the financial or medical field
- Professional approach to schedules, budgets, and quality of deliverables
- Basic knowledge of terminology management

Many translation schools and universities are now reviewing their curricula to meet industry requirements. In many cases, general computer training or localization classes have been introduced. Some universities, such as the University of Limerick in Ireland, offer postgraduate courses in localization, covering subjects such as language engineering, localization process, computer programming, technical communication, and quality control in localization. The Monterey Institute of International Studies in California ([www.miiis.edu](http://www.miiis.edu)) and Kent State University in Ohio ([appling.kent.edu](http://appling.kent.edu)) are training language and translation students in localization and project management.

Groups of industry and education specialists, such as LEIT and LETRAC, have researched available courseware and industry needs on internationalization, globalization, and localization.

LEIT is short for LISA Education Initiative Taskforce, a commission that was founded in March 1998. The commission consists of representatives from universities in the U.S. and Europe and has been commissioned to evaluate the status of localization-related education. In 1998, LEIT released the Industry Needs Survey report with facts and figures from a preliminary survey. The LEIT web site at [www.lisa.org/leit](http://www.lisa.org/leit) contains this report and a growing list of existing educational material and courseware.

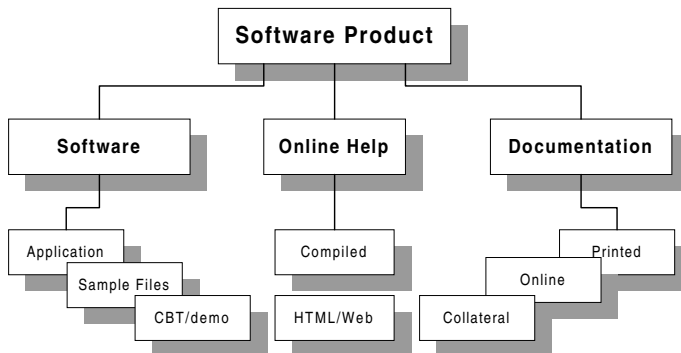
LETRAC is short for Language Engineering for Translators Curricula, a project funded by the European Commission, DG XIII, within the Telematics Application Programme of the Fourth Framework. LETRAC was an initiative that concluded in early 1999. The aim of the project was to “investigate the requirements and changing working environments of prospective employers of translators in both the private and public sectors to determine qualification requirements from an industry point of view.” For more information on the LETRAC project and an overview of the findings, visit the LETRAC web Site at [www.iai.uni-sb.de/LETRAC](http://www.iai.uni-sb.de/LETRAC).

## 4 Localization Projects

This section provides an overview of a typical software localization project, i.e. components, people involved, and the localization process. The text contains references to other sections in the book that cover certain steps in the process in more detail.

### 4.1 Project Components

Traditionally, localization projects involve the following components:



Today, localization projects increasingly include additional support material, such as web-based technical support data. In some cases, software products are partially localized for certain target markets. For example, a software publisher may decide to translate the software and not documentation, or only parts of the complete documentation set, based on sales and marketing requirements and return-on-

investment expectations. A common choice for non-desktop applications is to translate the installation guide and getting started guide, and to include administrator or reference manuals in English.

Below you will find more detailed descriptions for each of the project components shown in the chart.

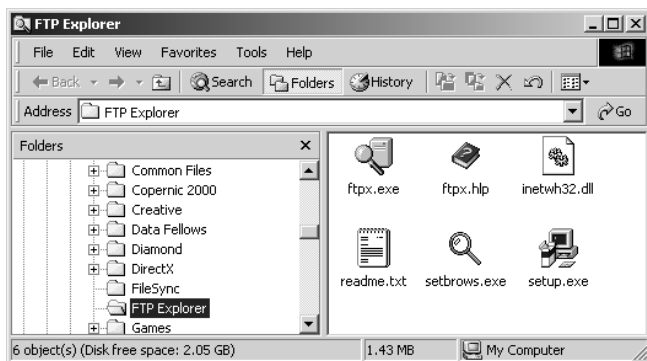
#### 4.1.1 SOFTWARE FILES

A software product is usually distributed on CD-ROM or through the Internet. Setup disks contain compressed installation files that are unpacked and copied to the hard drive by a setup program, or installer.

Most software applications install the following file types on a computer's hard drive:

- Program files
- Online help or online documentation files
- Readme files

On PCs with Microsoft Windows installed, program files typically have file extensions such as .exe, .dll, and .ocx. Online help files have extensions such as .chm (HTML Help) or .hlp (WinHelp), and online documentation files often are in Adobe Acrobat PDF format. Readme files usually have .txt, .wri or .doc extensions. On Apple Mac OS computers, files have no file extensions but internal file type definitions and custom icons.



Software publishers often include sample files, tutorials, demos, and trial versions of other products with their application. Some, or even all of these files may also require localization.

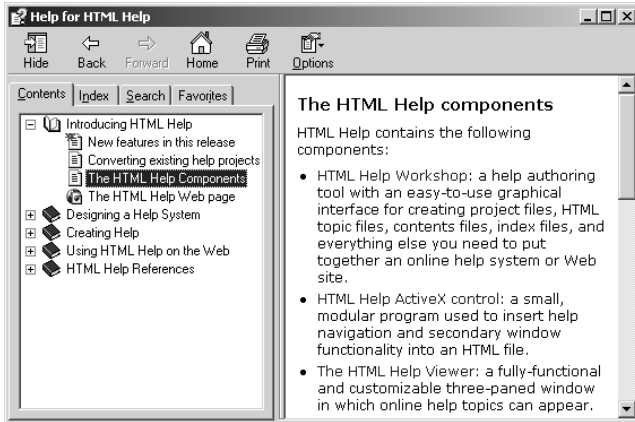
For more information on translating software, refer to Chapter 3, Software Translation.

#### 4.1.2 ONLINE HELP

Online help consists of documents that can be viewed on-screen and contain hypertext links that enable the user to jump from one topic to another. These documents can normally be accessed from within the application. Most online help systems are context-sensitive, which means that from within a particular section of the application

it is possible to directly access help information on that section, for example a dialog box or a particular option.

Increasingly, online help is distributed in HTML format. In the case of Microsoft's HTML Help standard, a set of HTML files is compiled into one central online help file that can be searched, printed, and annotated.



For more information on translating online help files, refer to Chapter 6, Online Help Translation.

### 4.1.3 DOCUMENTATION

In most cases, software publishers include online documentation with their products instead of printed documentation. This not only reduces printing costs, but also makes it easier to distribute updates or fixes to the documentation via the Internet.

Many shrink-wrapped software products come with only one or two printed manuals, typically an installation guide and a getting started guide. The installation guide tells the user how to install the application; getting started guides provide a short introduction to the application. Detailed procedural information can be found in the online help or online documentation.

Other types of printed documents that are often included with software packages are quick reference cards, marketing materials, and registration cards. These types of documents are often referred to as *collateral material*. Other examples of collaterals are CD labels and the product packaging.

Online manuals are usually provided in an electronic exchange format, such as Adobe's Portable Document Format (PDF). Online manuals can be opened and viewed using a helper application, which can also be used to print sections or pages. Adobe Acrobat Reader is the freeware viewer for PDF files.

The main difference between online help and online documentation is in their use. Online help is frequently accessed from within the application, can be searched, and enables users to jump between topics to find related information. Although many of these features may also be present in online documentation, it is less focused on on-

screen use. Much more important is that users can view the online documentation and print the pages they need.

For more information on translating (online) documentation, refer to Chapter 8, Documentation Translation.

#### 4.1.4 WEB SITES

Global enterprises often provide information on their web site in a number of languages, enabling visitors to choose the required language from the opening page. They usually not only require the company and product or service information translated, but also their entire product catalog, e-commerce, and support pages.

Today's web sites are often based on a database repository that publishes text and other information dynamically to HTML or ASP pages. With the introduction of XML, the use of databases to create web site content will become even more widespread. For web site localization this means that where traditionally large sets of HTML pages and images needed to be localized, now database tables with structured content will be translated. As a result, web site localization is more comparable to software localization than online help localization, i.e. extensive linguistic and functional testing will be required.

Another complex issue when localizing web sites is the speed at which content is updated; ideally all updates on multilingual web sites are published simultaneously in all target languages. For this reason, many web integrators and localization vendors are designing translation workflow solutions that transfer changed or updated content to vendors or individual translators and post the translations back to the web site.

Even though this book does not contain a chapter dedicated to web localization, most of the information provided is relevant to localizing web sites and web-based software applications, especially the HTML and XML sections in the online help chapters.

## 4.2 People Involved

Traditionally, any or all of the following parties are involved in a localization project:

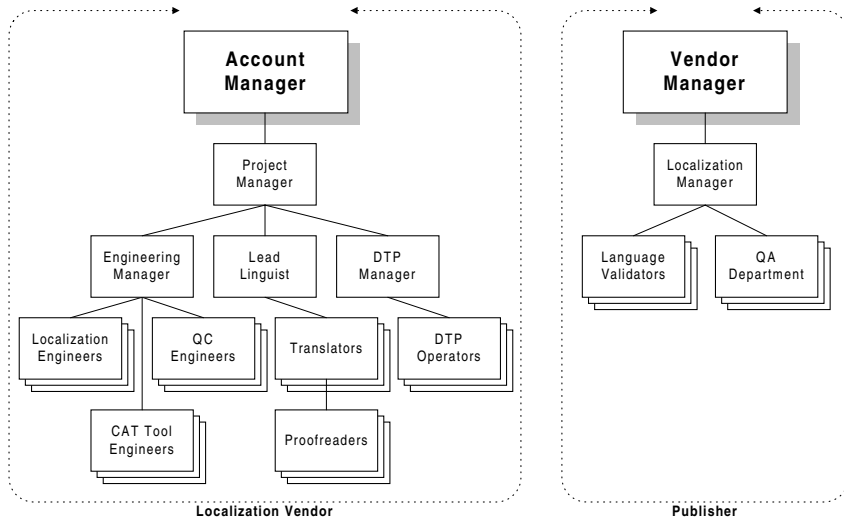
- The publisher, utilizing centralized in-house project management, engineering, and translation support staff
- The publisher's subsidiary or distributor in the target language country
- A localization service provider

Today, with companies globalizing and the web cutting out many of the middlemen, increasingly publishers are dealing directly with localization vendors to integrate localization in the overall development process. Many localization vendors are even setting up infrastructures that enable developers or publishers to interact directly with individual translators or single-language vendors in order to achieve simultaneous release of all language versions.

Software and web publishers normally outsource localization work to specialized localization and translation service providers to avoid setting up large in-house

localization groups that only have work for limited periods of time, e.g. the months before the release of a new product. In addition to translation and localization, in many cases, software publishers also outsource other activities to service providers, such as functional testing, document authoring and management, internationalization, or multilingual product support.

The following diagram displays a high-level overview of the people who are directly involved in localization projects, both on the localization vendor and publisher side.



#### 4.2.1 SOFTWARE PUBLISHERS

Large software publishers that localize their products in a number of languages usually employ one or more full-time vendor managers and localization managers, and may also have a dedicated quality assurance department. These people are responsible for managing, supporting, and monitoring the overall localization process as the work is produced by the localization vendors.

##### Vendor Manager

A vendor manager is responsible for all business relationships with localization service providers and works with vendors on much of the non-project-specific issues. The tasks of a vendor manager may include:

- Creating requests for quotation (RFQ) or requests for proposal (RFP) that are sent out to localization service providers prior to new projects.
- Developing long-term partnerships with strategic localization vendors.
- Visiting, assessing, and selecting vendors, based on specific internal quality and time to market goals and requirements.
- Negotiating prices and conditions with selected vendors.
- Supervising vendor contract administration.



- Monitoring the quality delivered by the selected vendors.
- Creating and monitoring localization budgets.

Small software publishers often combine the function of vendor manager and localization manager in one role. In many cases, this role is more advisory than decision making, but this does not apply to all organizations.

### Localization Manager

A localization manager typically coordinates the projects that are ongoing. The tasks of a localization manager include:

- Organizing kick-off meetings for new projects.
- Defining processes and contact points for localization vendors.
- Distributing project materials and guidelines to vendors.
- Overseeing overall project progress, budget, and contracts.
- Coordinating post-localization release activities, such as CD creation, printing, or electronic distribution.
- Assisting the vendor manager in selecting the right localization vendor.
- Managing or coordinating project teams on the vendor side.

Localization managers are sometimes called program managers or project managers. They are ultimately responsible for the timely localization of a product.

### Quality Assurance Department

The quality assurance department of the software publisher performs a final quality check on all localized components received from the localization service provider to look for problems or bugs in the language or functionality. The QA group also usually performs quality checks on samples during the project.

Depending on the degree of outsourcing that the software publisher has decided upon, localized versions of the product are tested by the internal QA department. However, software publishers increasingly also outsource full functional testing to the localization provider, to prevent bottlenecks just before all localized versions are to be released. In many cases, localization providers are required to hand-off gold-master versions of the localized product, which have been fully tested and are ready for release.

### In-country Language Reviewers

Most large publishers ask in-country reviewers working in their local offices, or product distributors to review samples of the files that have been localized by the localization service providers. This review or validation is not the same as editing or proofreading; reviewers must focus on technical consistency, completeness, and adherence to agreed terminology and language standards.

Often, these reviewers are in direct contact with the lead translators of the localization service provider to discuss linguistic issues.

## 4.2.2 LOCALIZATION SERVICE PROVIDERS

At localization service providers, the people involved in a typical project are:

### Account Manager

The account manager, sometimes called program manager, is the direct contact for publishers on all non-project related issues, such as contract negotiations, finances, general, and quality issues. In the case of serious project-related issues, account managers may get involved as well. Account managers are often also responsible for generating new leads and developing new business.

### Project Manager

Project managers schedule projects, assign resources, communicate with clients about project-related issues, and monitor the project progress and budget.

For more information on project management, refer to Chapter 14 of this book.

### Localization Specialist/Senior Translator/Translator

Localization specialists or senior translators review the work that in-house translators or freelance translators do. They also set style standards and manage terminology.

Often linguists who translate software applications are called “localizers” because they also get involved in other project activities such as software user interface resizing.

### Proofreader/QA Specialist

A dedicated proofreader or reviewer proofreads translated software, online help, and documentation files. A proofreader does not necessarily focus on the *translation* quality, but especially on the final *linguistic* quality of the product. Proofreading is a final language check for spelling errors, grammar mistakes, and consistency.

For more information on proofreading, refer to the Reviewing Documentation section in Chapter 8 of this book.

### Localization Engineer/Testing or QC Engineer

Localization engineers are responsible for all technical aspects of localization projects, including project preparation, software and online help engineering, compiling and testing. Often, dedicated testing or quality control engineers take responsibility for the final, functionality testing of a translated product.

Another task that is often performed by localization engineers is the creation of screen captures which are embedded in the localized online help or documentation files.

For more information on localization engineering and testing, refer to Chapters 4, 5, 7, and 10 of this book.

### CAT Tools Specialist

A Computer Aided Translation Tools Specialist is usually part of the engineering team, and specializes in the correct and efficient use of CAT tools such as translation memory and software localization tools.

Tasks performed by CAT Tools Specialists include extracting, importing, and processing of text using tools, selecting the right tools for specific file formats, and creating customized filters or parsers using scripting languages.

For more information on computer aided translation tools and other translation technology, refer to Chapter 11.

### DTP Operator

DTP operators take responsibility for the layout of the printed or online material and usually prepare localized files for pre-press production, or conversion of printed documentation to online format.

For more information on desktop publishing, refer to Chapter 9.

## 4.3 Localization Process

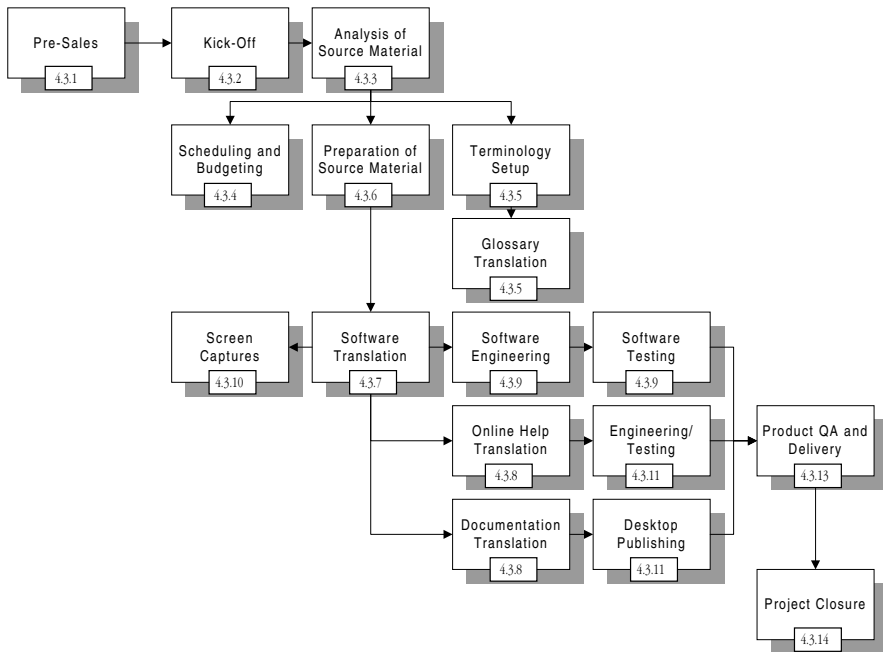
No two localization projects are the same. Each project has its surprises, in a positive or negative sense. Sometimes, projects that were announced as huge, technically complex operations are over before you know it. On the other hand, it is equally possible that small projects can turn into never-ending nightmares.

Each task in the localization process, including linguistic, technical, and project management steps, should be defined accurately before project initiation so the scope of the project is fully clear. Preparation is key in every project!

The workflow of a typical localization project can be summarized in the following steps:

1. Pre-Sales Phase
2. Kick-Off Meeting
3. Analysis of Source Material
4. Scheduling and Budgeting
5. Terminology Setup
6. Preparation of Source Material
7. Translation of Software
8. Translation of Online Help and Documentation
9. Engineering and Testing of Software
10. Screen Captures
11. Help Engineering and DTP of Documentation
12. Processing Updates
13. Product QA and Delivery
14. Project Closure

Please note that the order of these steps can differ substantially depending on the project. Besides, two or more tasks can run simultaneously.



The following sections provide a more detailed overview of these steps and other issues that need to be taken into consideration during localization projects.

### 4.3.1 PRE-SALES PHASE

Before a project is awarded to a localization vendor, the vendor manager of the software publisher usually sends out requests for quotation (RFQ) or requests for proposal (RFP) to multiple vendors in order to get the most competitive bid.

RFQs are usually accompanied by a CD containing source material in order to enable localization vendors to create a realistic quotation and project proposal. The vendor evaluates the source material and then provides a quotation with a breakdown of all project activities and volumes. For more information on evaluating source material, refer to Chapter 13, Project Evaluations.

For more information on RFQs and creating quotations, refer to Chapter 14, Project Management.

### 4.3.2 KICK-OFF MEETING

Once the project has been awarded to a vendor, software publishers often organize kick-off meetings for translators, localization engineers and project managers for product training, to provide an overview of the project, to explain the localization procedures and methods, and – last but not least – to meet.

### 4.3.3 ANALYSIS OF SOURCE MATERIAL

When files for a localization project are received from a publisher, they first need to be analyzed by production specialists, i.e. engineers for the software and online help components, and DTP operators for documentation. Source files are mostly sent directly from a publisher's development department in a localization kit that includes the following:

- Running version of the English product
- Build environment for the software including source files
- Build environment for the online help including source files
- Documentation folder containing all book files, chapter files and graphics
- If appropriate, previously translated versions of the product or existing glossaries or translation memories

Larger localization vendors have a project evaluation team in place that analyzes all aspects of a new localization project, identifies problem areas, selects tools and approach, and provides the project managers with all information necessary to create a project schedule, budget, and resourcing plan.

Examples of information required by project managers are the project scope, the volumes, the recommended localization approach, and additional investments that may be necessary.

For more information, refer to Chapter 13, Project Evaluations.

### 4.3.4 SCHEDULING AND BUDGETING

Scheduling is a critical task in localization. Many issues endanger the time schedule set at the beginning of a project, the most likely one being late delivery of the source material by the publisher. Because many software publishers are aiming for simultaneous release – also called *simship* – of all language versions of their product, most localization vendors experience heavy workloads towards the end of a project.

For more information on scheduling localization projects, refer to the Scheduling section on page 443.

### 4.3.5 TERMINOLOGY SETUP

Most localization projects start with the creation of a multilingual product-specific glossary, which should ideally be approved by the publisher before any localization work is done.

This basic terminology list, also called project glossary, would typically contain terms that are commonly used in the product user interface or support documentation. A project glossary would also clearly indicate which essential terms, such as product names, should *not* be localized. For more information on glossaries, refer to the Project Glossaries section on page 403.

Please note that in professional terminology management, source terms are not just *translated*, but multilingual equivalents are established by research. In fact, translated

terms are only valid when the translations are found in several reference documents in the target language; obviously the web is a great resource for terminology research.

To obtain accurate multilingual equivalents from the very beginning of the project, provide terminology researchers with sufficient reference material, such as the running application or online help file. Translating key terms or finding equivalent terms in the target language without any contextual information on the product may result in major terminology changes later in the project.

In most cases, the project glossary is sent to the publisher for review and approval. After validation of the glossary, translation of the project's source material can start. During the project, the terminology database should be maintained and updated.

For more information on Terminology Management, refer to Chapter 12, Terminology.

#### 4.3.6 PREPARATION OF SOURCE MATERIAL

When the engineers and DTP operators have analyzed the source material, spotted potential problems and test-compiled all source material, a translation kit is created for the translators. Especially for multilingual projects, project preparation is a vital step because problems caused by insufficient preparation are multiplied by the number of target languages of the project.

Preparing a translation kit includes investigating leveraging possibilities for the software, i.e. checking whether existing translations can be automatically re-used. If previous translations of the software resource files exist, the localization engineer should try to re-use these translations as much as possible to avoid unnecessary work for the translators. This not only improves consistency, it also decreases the number of words to translate, i.e. speeds up the software translation process and reduces the cost for the publisher. For more information on the preparation of software files for translation, refer to Chapter 4, Software Engineering.

The same issues apply to online help and documentation files. If translation memories exist, they need to be included with the localization kit, along with files that have been pre-processed for use in a translation memory tool. If no translation memories exist, but an earlier version of the online help or documentation files has been translated, the possibility of *alignment* needs to be considered. Chapter 11, Translation Technology, provides more information on aligning translated text to create translation memory databases.

For more information on the preparation of online help and documentation files for translation, refer to Chapter 7, Online Help Engineering and Testing, and Chapter 9, Desktop Publishing.

The technical and linguistic information should be combined into one coherent set of instructions that is given to the translator along with all source material.

#### 4.3.7 TRANSLATION OF SOFTWARE

Ideally, software is translated first in localization projects. Translators edit resources in text-only files, in a resource editor, a translation memory tool, or in a software localization tool. For more information on translating software files, refer to Chapter 3, Software Translation.

The recommended order in which to translate software application resources is: dialog boxes, menus and strings. Usually, the string section contains most of the translatable text, i.e. error messages, status messages, and help text.

Once dialog box resources and menus have been translated, it is possible to create a preliminary user interface glossary – or even a preliminary software build – of these translations, which can be used as a reference point for translators of online help and documentation. Likely, most of the software references in help and documentation contain dialog box items and menu names, so it is not recommended to start translation of help and documentation before these items have been localized. For more information on creating software glossaries, refer to the Creating User Interface Glossaries section on page 409.

Once all resource files have been translated, the software application is compiled and dialog boxes resized. For more information on compiling and resizing localized software applications, refer to Chapter 4, Software Engineering.

Next, a linguistic test is performed to validate the translations in context in the running application. Once linguistic testing of the localized software is completed, the terminology of the application should be final and frozen. This means that a localized software version can be sent to the client for an acceptance test. In addition, vendors can now start creating screen captures for inclusion in the online help and documentation. For more information on creating screen captures, refer to the Creating Screen Captures section on page 349.

#### **4.3.8 TRANSLATION OF ONLINE HELP AND DOCUMENTATION**

As soon as a software glossary or a preliminary build of the localized software is available, translation of online help and documentation can start.

For more information on translating online help and documentation, refer to Chapter 6, Online Help Translation, and Chapter 8, Documentation Translation.

#### **4.3.9 ENGINEERING AND TESTING OF SOFTWARE**

Engineering localized software primarily involves resizing the user interface, assigning unique hot keys, and compiling the localized resource files into a running application. For more information on engineering localized software applications, refer to Chapter 4, Software Engineering.

Many different types of testing are possible in a localization project. The first testing type, linguistic testing, is often seen as part of the translation process and should be done by a translator or linguist, if possible with the help of a localization engineer or test scripts.

If included in the statement of work for the project, functionality testing is performed by localization engineers or dedicated testing and QC teams. Often the publisher provides test scripts, which guide engineers through the testing process. If no guidelines or scripts are provided, ask the software publisher how much time should be scheduled for functionality or compatibility testing of the localized versions of the software. Typically, a subset of the test scripts used for the English version is created to test the functionality of the localized versions.

Refer to Chapter 5, Software Quality Assurance, for more information on testing localized software applications.

#### **4.3.10 SCREEN CAPTURES**

Screen captures are pictures of user interface elements of the running software that are used in online help or documentation to show a task situation or context, the objects being described, or the result of performing a procedural step. For example, a screen capture can show a dialog box with certain options selected.

Screen captures need to be created for each target language. Without screen captures of the localized software, no online help file or manual can be finalized. Therefore, it is recommended to start creating screen captures as soon as the linguistic testing of the software is completed and the user interface terminology has been frozen.

For more information on creating screen captures, refer to the Creating Screen Captures section on page 349.

#### **4.3.11 HELP ENGINEERING AND DTP OF DOCUMENTATION**

As soon as all graphics and screen captures have been localized and reviewed, desktop publishing of documentation and testing of online help projects can start. Although it is possible to perform DTP and online help testing when the graphics have not been localized yet, using the original graphics, this will require an additional check after the graphics have been translated. For example, image sizes may change in localized versions because of resizing of dialog boxes. As a result, page flows may change so DTP work needs to be re-done.

For more information on engineering and testing online help files, refer to Chapter 7, Online Help Engineering and Testing. For more information on desktop publishing, refer to Chapter 9, Desktop Publishing.

#### **4.3.12 PROCESSING UPDATES**

Translation often starts when the English product is still being developed. This means that first one or more beta versions or release candidates of the product are translated before the final version is released to localization.

If vendors are not using a translation memory system for translation, they need to manually process updates using file compares, copying and pasting, etc. Translation memory systems enable the user to process new files and automatically re-use the translations that were already completed. Refer to Chapter 11, Translation Technology, for more information on the use of translation memory tools.

When desktop publishing for printed documentation or testing of online help has already started, it is more difficult to process updates. After translation in a translation memory tool, files are converted back to their native formats, and after that point additional edits or corrections are no longer stored in translation memory. In case of these last-minute updates, either process the updates using the translation memory tool and lose all desktop publishing or testing work, or insert the changes in the files that are being worked on. In the latter case, the translation memory will not be up-to-date anymore, unless it is updated separately.



Most of the time, there will be too much time pressure to re-do desktop publishing and testing work, and the latter procedure will be chosen.

To prevent unnecessary software engineering, testing, or desktop publishing work, project managers should always agree with their clients if – and when – updates can be expected.

#### 4.3.13 PRODUCT QA AND DELIVERY

Before delivery, a quality assurance check should be performed on all translated material. At this stage, the software publisher may also ask to see the files for sign-off.

This pre-delivery QA check includes:

- Proofreading all translations. For more information on proofreading, refer to the Reviewing Documentation section on page 315.
- Delivery test of the software. For more information on this type of testing, refer to the Delivery Testing section on page 154.
- Finalizing bug or problem reports. For more information on bug reports, refer to the Bug Tracking section on page 160.
- Reviewing the instructions given in the initial hand-off or statement of work from the publisher to ensure that all steps were covered.

When the QA checks have been completed, the translated material can be delivered to the publisher. After all translated files have been delivered and approved by the publisher, all project materials are archived.

#### 4.3.14 PROJECT CLOSURE

Many publishers organize a post-mortem – or project audit or wrap-up meeting – with the localization vendor after a project has been completed. Issues discussed in these meetings include:

- Process-oriented evaluation of the completed project
- Evaluation of technical and linguistic quality of deliverables
- Identification of areas for improvement
- Suggestions for process modifications for future projects

Please note that these meetings work both ways; both parties need to provide input on each other's performance.

During the post-mortem meeting, one of the attendees should take notes and distribute a summary of all topics discussed to both parties. This document could function as a starting point for new projects.

## 5 Further Reading

Not many books have been written covering software localization or the localization industry. The following references cover different aspects of the localization industry. For example, Robert Sprung's book deals with business case studies of localization vendors and publishers, and Hickey's book is a collection of daily job and task descriptions of various people working for a localization vendor, from project managers to translators and engineers.

Hickey, Tim. ed. 1999. *The Guide to Product Translation and Localization*. Computer Society, ISBN 0-769-50022-6.

Hoft, Nancy. 1995. *International Technical Communication*. John Wiley & Sons, ISBN 0-471-03743-5.

Sprung, Robert. ed. 2000. *Translating Into Success: Case Studies in Translation, Localization, and Internationalization*. John Benjamins, ISBN 1-556-19631-8.

# Chapter 2: Internationalization

This chapter is aimed at software publishers or web developers who want to learn more about the internationalization of their products. The chapter provides an overview on internationalization and further relevant information for managers, software programmers, web developers, and technical writers.

This chapter includes the following sections:

1. [Introduction](#)
2. [Language](#)
3. [Software](#)
4. [Web Sites and HTML](#)
5. [Documentation](#)
6. [Creating Localization Kits](#)
7. [Further Reading](#)

## 1 Introduction

The localization of a product does not begin with the delivery of source material to the localization vendor. Publishers 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.

As mentioned in the introduction, the Localisation Industry Standards Association (LISA) defines internationalization as follows:

[“Internationalization is the process of generalizing a product so that it can handle multiple languages and cultural conventions without the need for re-design. Internationalization takes place at the level of program design and document development.”](#)

There are two key reasons for internationalizing software and (online) information:

1. To ensure that a product is functional and accepted in international markets.
2. To ensure that a product is localizable.

In the first case, the product is designed to support features such as international character sets, keyboard layouts, date and time formats, and currencies. The online help 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.

The second reason above will help to reduce localization costs by developing the product in a way that ensures a smooth localization process, both for the software publisher and the localization vendor. One way to do this in the software is to externalize all translatable components of a software application to make them easy to identify and translate. In the online help and documentation, using generic country-independent examples, or creating multi-layered images reduces costs considerably.

Here is an example of problems that may be prevented by internationalizing products during development:

- Incorrect display of accented characters. If the product has no built-in support for character sets that fall outside the range of the standard ASCII set, accented characters may display incorrectly or garbled. Unicode support in a product guarantees support, i.e. correct processing and display, for all of the world's language scripts.
- Text truncations. Translated text is normally about 30% longer (sentences) than the English original, but the increase may be as much as 100 percent (single words). If there are significant length restrictions in the user interface of a software application, translators will have to abbreviate items or choose synonyms, which may affect the quality of the localized product or confuse the end user. If this is the case, the user interface may have to undergo extensive resizing, which will increase production time and costs. This is a particular problem in ideographic languages, such as Japanese or Chinese, where abbreviations are not used and vertical expansion may also be required.



- Fixed date, time, number, or currency formats. Apart from different separators, e.g. 1/6/2001 in the United States and 1.6.2001 in German, the order is also important. For example, 6/10/2001 means 10 June 2001 in the United States and 6 October 2001 in European countries.
- U.S. specific icons or graphics. Icons may have different meanings in different countries. When Apple first used the Trashcan icon, it was interpreted as a mail box to European Mac OS users.

“Enablement” is a term that is commonly used in the context of internationalization. It refers to the process of adjusting software to make it functional in certain countries or geographical regions. For example, a software product can be “double-byte enabled” to process and display text in Asian languages.

Internationalization can be applied to the online help and documentation, as well as software. Here are some examples of internationalization issues in documentation:

- Jargon or U.S. specific language. Many internationalization problems have been found in sample files, such as document templates, which were not designed with international users in mind.
- Manually generated table of contents or index. If the source material does not contain an automatically generated table of contents or index, these elements need to be recreated manually for each target language. Getting the index page numbers right is a very time-consuming process, and leaves a considerable margin for errors. Automating this process will help avoid any problems.
- Text in graphics. Adding large chunks of text in icons or bitmaps will increase localization costs and add to the translation and editing time for graphics. Ideally, the level of text in graphics should be kept to a minimum, and should be highlighted at the internationalization phase and the implications understood.

The following sections describe some of the methods used to internationalize a software product, along with examples, check lists, and tools that may simplify the internationalization effort.

## 2 Language

To date, the majority of software products have been developed in the United States. As a result, the bulk of software localization work is performed on English source language products for other languages. Influencing the way language is used in the English product can have a great impact on the acceptance and localizability of the product.

Typical examples of product components that often pose difficulties during the translation cycle are sample files and template documents. With these types of documents, it is advisable to use generic examples and models which can be used internationally with little or no modification.

### 2.1 Overview

Most software publishers employ a group of technical writers and authors who write and develop the online help and documentation text. It is not an easy task for writers whose native language is English to “write globally”, i.e. use examples that can be understood world-wide, because this will often limit the type of examples they can use and the style they can adopt.

The most important thing in a written text is that the text must be understood by non-native readers. The terminology and style used should, therefore, be basic and uniform. Furthermore, the text must be written with translation in mind, so the translator can work quickly and accurately, without the need for clarifications, rewrites, or cultural modifications. Typical examples of a writing style that could make the translation work

easier are short sentences, simple vocabulary, consistency in terminology, and careful use of punctuation.

Moreover, a structural consistency in the source files greatly improves leveraging of information within the same files and also across file sets, especially where CAT tools are used in the translation process.

## 2.2 Check List

Ensure technical writers keep the following issues in mind:

### 2.2.1 TERMINOLOGY

- ✓ Create a glossary of terms relating to the product, company or industry, and apply this consistently across all documentation and online help sets.
- ✓ Use consistent phrases and terminology. The importance of simple, concise language is magnified when writing for translation. For example, decide at the outset if you want to use phrases like “click on”, “click”, “choose”, or “select”, when describing software commands.
- ✓ Maintain consistency between the documentation or online help text and the software user interface.

### 2.2.2 STYLE

- ✓ Ensure that the text is clear and unambiguous, for example refrain from using verbs like “may” or “might”; use “can” instead.
- ✓ Avoid the use of “telegraphic English”, i.e. write full sentences.
- ✓ Use short, concise sentences, preferably in active voice. For example, “Choose the Copy command” is much clearer than “the Copy command should now be chosen.”
- ✓ Do not use culturally-specific text or jargon, e.g. humor, political references, slang, references to TV shows, national monuments, sports equipment, hand gestures, or ethnic stereotypes. Equivalents in other languages can probably be found, but this requires much more effort than straight translation and increases costs.
- ✓ Avoid references to or images of animals. In some Asian regions, for example, this may be considered offensive. Be careful in using references to religions, sacred objects, or symbols.
- ✓ Avoid references to seasons, time zones, weather, or holidays, such as Christmas trees or Halloween pumpkins.
- ✓ Avoid using acronyms and abbreviations, e.g. NBA (National Basketball Association). Abbreviations are often difficult to reproduce in other languages. If acronyms have to be used, spell them out in the first occurrence in the text.
- ✓ Avoid using long modifier chains or ambiguous phrases, such as “required disk slot”. This phrase can be interpreted as “the slot for a required disk” or a “disk slot which is required”.

- ✓ Consider measurement conversion. For example, consider adding “(2.54 cm)” when referring to “1 inch”. This will also avoid inconsistencies between the conversions done in different target languages; some translators may leave the “1 inch” unchanged, others may completely replace it with the conversion. Anticipate any confusion.
- ✓ In marketing material, avoid comparative statements on competitor’s products. In many countries outside the U.S. this is unusual or even illegal.

### 2.2.3 EXPANSION

- ✓ Leave sufficient space for text expansion in tables, flow charts, and illustrations, i.e. anticipate text expansion, particularly in software user interfaces.
- ✓ When text is localized into double-byte languages, anticipate vertical expansion.
- ✓ Use only one space between sentences and after all punctuation marks.

### 2.2.4 COUNTRY-SPECIFIC ELEMENTS

- ✓ Write dates out fully, e.g. write “6 November 2000” instead of “6/11/2000” because the month/date order varies in some languages.
- ✓ Be aware that many countries have different calendars, e.g. in India and Thailand the calendar starts from Buddha’s birth.
- ✓ In telephone numbers and addresses, always include the country code and country name. Specify the hours and time zone when customer support services are available.
- ✓ When including examples such as a person’s name, use generic examples that are known world-wide.
- ✓ Check if the warranty and technical support text is relevant and valid in the target countries. If not, consider having them translated or reviewed by legal in-country staff or consultants.

Some of the items listed above are standard issues for technical documentation authoring, nevertheless it is advisable to list all items in an internal, international style guide for the authoring team. This style guide could be based on standard works like the *Chicago Manual of Style* or the *Microsoft Manual of Style*.

## 2.3 Controlled Language

Controlling or structuring the use of language is an important factor in reducing the costs of writing online help and documentation, and the time needed to develop and update it. Controlled language, controlled English, easy English, structured authoring, or simplified English is a writing system or method which helps technical authors avoid ambiguity, and improve the readability of their text. There is no single standard for controlled language; several systems have been developed and applied with varying degrees of success.

A controlled language is a subset of a natural language and is used to write technical documentation, usually for a particular company or industry sector. Its basic

components can include a set of approved terms for certain concepts, a list of words and phrases to be avoided, along with their approved synonyms, and a set of grammar rules.

Using controlled language has the following advantages:

- The information presented can only be interpreted in one way, so ambiguity is reduced and readability improved.
- Consistent use of style and terminology is enforced.
- The translatability of the text is improved, especially when computer-assisted translation tools, e.g. machine translation, are used in the localization process.
- The time to market for localized products can be shortened because translation speed will be increased, often up to 25%.

Several companies have developed software applications that assist writers in using controlled language. These can be run in conjunction with the editor used by technical writers. Most of these tools work either in batch mode or interactive mode, presenting grammar and vocabulary suggestions in an external word processor or desktop publishing application. The tools basically place constraints on the vocabulary and syntactic structures that an author can use.

Even though the resulting documents may be easier to read and understand, they are not necessarily easier to write. A few examples of these constraints include a maximum or restricted number of words per sentence, no passive clauses, restrictions on the use of tenses that may be difficult to translate, and a defined terminology domain.

Controlled language has been used for years in the aerospace and automotive industries, and is gradually finding its way in the IT industry, often combined with SGML- or XML-based content creation and document management systems. One of the first companies to develop a controlled English, called Fundamental English, was Caterpillar Tractor Corporation. In the aerospace industry, the AECMA Simplified English standard was released in 1986. Both systems outline style and grammar rules and list both allowed and proscribed vocabulary.

Here are some examples of commercially available tools for controlled language writing:

- ClearCheck by Logica Carnegie Group – more information at [www.logica.com](http://www.logica.com) or [www.cgi.com](http://www.cgi.com)
- LANT@MASTER by LANT – [www.lant.com](http://www.lant.com)
- MAXit by SMART Communications – [www.smartny.com](http://www.smartny.com)

Further information on Controlled English can be found at [www.eamt.org](http://www.eamt.org).



## 3 Software

Internationalization of software code has always been a complex task. More often than not, software products have been developed without any consideration for localization issues. This makes it very difficult to re-design the code for a localized version.

The best approach to take when developing software is to maintain language independence throughout the entire programming code. No portion of the software should contain code that is specific to any one language. Product development and internationalization should be treated as a single, integrated process. Moreover, the code should rely on language support offered by the operating system or another qualified language application programming interface (API).

Most development environment manuals contain chapters on developing software for international markets. Microsoft, for example, includes chapters on internationalization in their Visual Studio documentation. This can be accessed through the Microsoft MSDN web site at [msdn.microsoft.com](http://msdn.microsoft.com) (search for “internationalization”).

### 3.1 Overview

Software publishers have to deal with several design issues when internationalizing software code for foreign markets. The most important issues are listed below:

1. Character encoding
2. Location of *translatables*, e.g. user interface text
3. User interface design
4. Regional standards

In the following sections, tips and resources are provided that may help you in internationalizing a software product. Remember that re-engineering a product after development is more time-consuming and problematic than addressing internationalization issues from the very beginning!

#### 3.1.1 CHARACTER ENCODING

Apart from having the capability to *display* all foreign characters in the software user interface, users should also be able to *input* international data in a software application, using a localized keyboard or other preferred input method.

The best advice for software publishers who want to enable their products for international markets is: Go Unicode! Unicode is a fixed-width, 16-bit character set, which means it can represent more than 65,000 characters. The Unicode standard is destined to replace the ASCII and other single- and multi-byte character sets currently in existence. Unicode encompasses scripts and general-purpose symbols for writing text in nearly every language.

If a software publisher considers localizing products for Far Eastern or Middle Eastern markets, making the source code and application Unicode compliant is worth the investment. Unicode is already supported by most current operating systems, including

Mac OS and Windows. The main advantages of Unicode support are the ability to represent all text with one unambiguous encoding, and the ease with which multilingual text can be supported.

For more information on Unicode, visit the Unicode Consortium web site at [www.unicode.org](http://www.unicode.org).

### 3.1.2 LOCATION OF TRANSLATABLES

For the location of translatables in source code, there are several approaches, the most common of which is centralizing all localizable items in external resource files.

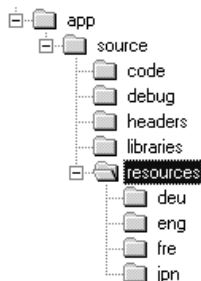
According to Microsoft's development guidelines, the most efficient way to design a software application is to split the application into two conceptual blocks: a code block and a data block. The data block contains all user interface items and no programming code, which means it is the only element that requires localization. All information from the data block can be centralized in a resource file which compiles into a resource-only language .dll. If a multilingual user interface is required, for example a button for switching between languages, name the resource-only .dll files with an easily identifiable language code, e.g. `res_fr.dll` for the French version.

Externalizing all translatables to one resource file has the following advantages:

- Efficiency – for each new language version of the application, only the resource file needs to be adapted.
- Security – localizers or vendors do not need to access the source code.
- Quality – reduces the chance of missing translatable strings, and limits the level of functional testing required.

For translators or localization vendors, dealing with source code that contains translatable text can pose some challenges, particularly where code and translatable text are indistinguishable. A string like "COPY" could be a status bar string, but could also be an internal command that would corrupt the application when changed.

In addition to externalizing all localizable strings to one file, it is advisable to create a build environment that is truly multilingual. Here is an example of a folder structure for a multilingual software build environment:



If separate folders containing language-specific resource files are used, create different project files to compile each language version.

### 3.1.3 USER INTERFACE

There are many things to consider when developing a software user interface. The most important thing is to allow for text expansion. Add approximately 30% extra space to each control or button in the dialog boxes. Ensure the menu tree fits on all screen resolutions. For example, the German translation for the Edit menu is “Bearbeiten”, and “Wiedergabe” for the View menu. In both cases, the text has been expanded by 100%. For this reason, many Windows applications in European languages use a question mark symbol (?) instead of the word “Help” on menu bars.

### 3.1.4 REGIONAL STANDARDS

Software applications should, of course, support all regional standards of the countries or regions where they are being shipped. Here are some examples of regional standards that should be supported by applications:

- Sorting rules, e.g. in Traditional Chinese sorting is based on the number of strokes in the character. In Swedish the letter Å sorts after Z, not near A.
- Measurement formats, e.g. inches in the U.S. versus centimeters in Europe
- Number formats, e.g. decimal separators
- Time/date formats
- Calendar formats
- Currency formats
- Address formats
- Phone numbers

Most development environments support the *locale* model, i.e. a collection of standard settings, rules and data specific to a language and geographical region.

## 3.2 Check List

Here is a check list for foreign character support:

- ✓ Ensure the product supports international sorting standards.
- ✓ Ensure the product is multi-byte enabled, for example by using Unicode.
- ✓ Verify that the users can successfully cut/paste text that contains accented (or double-byte) characters to other applications.
- ✓ Verify that the application can open and save files with accented characters.
- ✓ Ensure the product supports international code sets, and supports both input and output (display) of accented characters.

Here is a checklist for location of translatables:

- ✓ Keep track of all resource files and *include files* (bitmaps, icons) to be localized.
- ✓ Avoid using too many separate, small resource files. Ideally all translatable text should be centralized in one or two files.

- ✓ Use variable names that are not “real” words to avoid translation of non-translatable strings.
- ✓ Use comments with code to provide context for translators.
- ✓ Exclude all non-translatable text strings from the resource files, unless they include comment text for translators.

Here is a checklist for the user interface 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.
- ✓ Use text-wrapping features instead of manual wrapping using `\n` control codes.
- ✓ If possible, restrict the use of “dynamic” dialog boxes, i.e. dialog boxes with options where text is retrieved from the strings section and dynamically inserted.
- ✓ Avoid concatenating strings, i.e. combining strings at run-time to form new strings. Composite messages are often very hard to translate.
- ✓ Avoid hardcoding keyboard commands.
- ✓ Make sure shortcut key combinations are accessible from international keyboards, e.g. use Function keys instead of letters in shortcut key combinations, e.g. `Ctrl+F3`.
- ✓ Delete any obsolete text or strings from the resource files.
- ✓ Avoid hardcoding coordinates of dialog box or control sizes.
- ✓ Avoid using placeholders or variables, such as `%s`, or include a comment line explaining to the localizer what will be inserted at run-time. Never add grammatical endings (such as the plural “s”) to variables.
- ✓ Allow localizers to change the font and font size of options in dialog boxes.
- ✓ Ensure custom, third-party controls in dialog boxes are localizable.
- ✓ Be cautious about using icons or bitmaps showing people, hand gestures, holiday symbols, or body language. Use generic images that do not have to be localized or adjusted for target locales.
- ✓ Avoid using colors or color combinations associated with national flags or political movements.
- ✓ Avoid using fixed variable orders.
- ✓ Limit the use of text or letters in bitmaps and icons.
- ✓ When variables are used within a string, add at least one extra line to text boxes.

Here is a checklist for locale support:

- ✓ Build in support in for DBCS (Double-Byte Character Sets) for Asian languages, i.e. 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 Asian languages.

- ✓ Use language support from the operating system running the software product.
- ✓ 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.

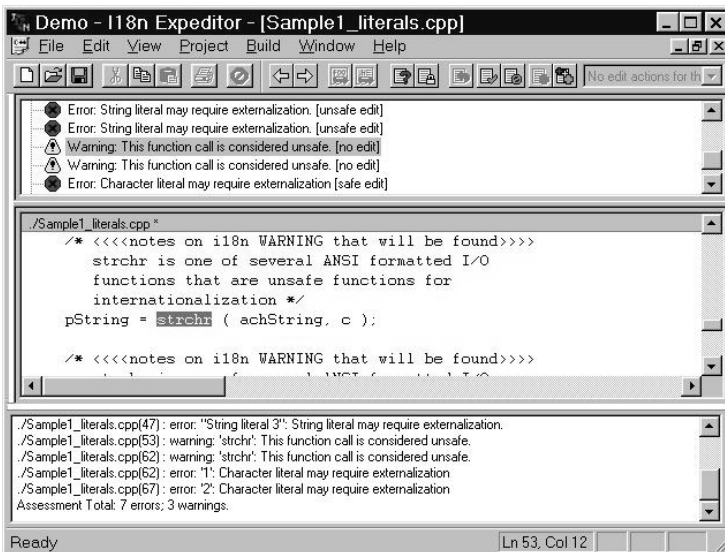
## 3.3 Tools

Several tools have been developed to help developers determine how well their products have been internationalized.

Two types of internationalization tools are available: code analyzers and software that modifies the code to support different locales and character sets using international libraries. The following sections provide some examples of these tools.

### 3.3.1 I18N EXPEDITOR

I18n Expeditor analyzes source code (C/C++ and Java) before compilation to detect and (often automatically) correct internationalization errors, such as hard-coded strings and character literals.



For more information on I18n Expeditor, visit the OneRealm web site at [www.onerealm.com](http://www.onerealm.com).

### 3.3.2 GLOBAL CHECKER

Uniscap's Global Checker is a software scanning utility that identifies issues with internationalization and provides solutions. It scans Java, C or C++ type source files and searches for language support issues based on the language and platform specified.

Uniscape's Global C provides programmers with a basic set of routines and functions providing platform-independent internationalization services, such as Unicode API's, message catalogs, and a locale definition utility.

For more information on Global Checker and Global C, visit the Uniscape web site at [www.uniscape.com](http://www.uniscape.com).

### 3.3.3 ALIS BATAM

Alis Batam offers instant, global language support to an existing Windows application. It creates an open, multilingual application, unrestricted by the Windows version it runs on, as input, character sets and display are all handled by the internationalization library and include full Unicode support.

For more information on Batam, visit the Alis web site at [www.alis.com](http://www.alis.com).

## 4 Web Sites and HTML

Like software, web sites and HTML documents should be designed and developed with foreign language support in mind. Internationalization of web sites implies developing the site content to simplify hosting and display of localized information. This applies to graphics, placement of text, and local content.

Because web site internationalization is very much integrated with localization, the process of publishing web sites in foreign languages is usually referred to as web site globalization. A well globalized and internationalized web site allows localization vendors to focus on translating the content, as opposed to processing a wide range of complex file formats and types.

In short, the web site globalization process can be summarized as follows:

1. Choose a web site platform, server, and software palette that supports multilingual site hosting.
2. Evaluate the site architecture to determine what should be changed to facilitate the localization process.
3. Identify which parts of the site can be translated and which parts must be re-written specific for a target country – local content versus global content.
4. Create a multilingual web site structure or content database with language folders and navigation system.
5. Choose a toolset, for example combining a multilingual content management system with tracking tools and translation memory.
6. Develop a site update and maintenance plan.

After these initial internationalization steps, select a localization vendor to translate the web site content, or create local content. If applicable, involve your local or regional offices in the content creation or review process for their locale.

## 4.1 Overview

It is not unusual to see web sites changing frequently, both from a content and design perspective. Web pages are often generated dynamically using templates, scripts, and databases. Maintaining web sites in several languages can be a tedious process if it is not well thought-out from the beginning.

For more information on multilingual web site development, visit the Localization and Internationalization web page from the World Wide Web Consortium at [www.w3c.org/international](http://www.w3c.org/international). Other good references for web site internationalization and globalization are the Worldpoint web site at [www.worldpoint.com](http://www.worldpoint.com), the Idiom web site at [www.idiominc.com](http://www.idiominc.com), Mark Bishop's book *How to Build a Successful International Web Site*, and the [www.multilingualwebmaster.com](http://www.multilingualwebmaster.com) web site.

### 4.1.1 WEB SITE DESIGN

When planning a web site, anticipate localization even if the initial version will be in one language only.

First of all, determine which pages will be language-specific, e.g. product descriptions, and which ones will be generic, e.g. product graphics or icons, and build a multilingual folder structure for the web site based on specific language folders. At this point it is important to decide whether it is preferable to have one central web site running a content database that contains all language versions of web site, or separate web sites for each target language or region. The latter is an option often favored by larger companies with a significant physical presence in each country.

Maintaining one central, multilingual web site is the most common solution for companies that maintain a relatively small web site or do not publish in a large number of languages. For example, the high-level folder structure of a centralized, multilingual web site may be set up as follows:

```
index.html
\us
\us\images
\fr
\fr\images
\de
\de\images
\shared images
```

The `index.html` home page contains links to all language folders and contains a language selection menu. The `\us`, `\fr`, and `\de` folders contain the English, French, and German files. Each language folder has its `\images` folder to store localized images, and there is a `\shared images` folder containing non-localized art, such as logos or bullet symbols.

It is also very important to consider a multilingual navigation system that enables users to select their language or country first, and then guides them through all pages in their native language. Ideally, the system should track the visitor's language preference, for example by leaving a cookie on the local machine. It is advisable to include a language choice button on every page, because users may access the page through a search

engine instead of the site's home page. List the languages in a selection menu with localized language names.



Many web sites today are database-driven or have a transaction processor, i.e. so-called dynamic web sites. Ensure that the database and platform chosen support content in foreign languages, particularly if there is a plan to localize into multi-byte languages. Check if common internationalization issues such as currency conversions and date or time formats are supported. A good way to verify language support is to run a pseudo-translation of the web site, including accented characters in all strings, or increasing the string length using a search and replace tool.

A properly internationalized web site has the following features:

- Support for non-Roman languages and character encodings, for example through the use of Unicode
- A scalable design which makes the localization process more efficient, for example, by limiting the use of text within graphics
- Clear separation of code and content, for example, by including all interface text in one central content database, or using cascading style sheets for sets of HTML pages
- Web page layout which is determined by language-specific style sheets or templates

A web site ready for localization would be Unicode compliant, stores all content in a database format, and publishes information dynamically using internationalized HTML templates and style sheets that allow for text expansion. Ideally, the site should have a tracking tool which would automatically detect changed or new content and send it directly to the assigned localization vendor and/or reviewers before it is returned and inserted in the content database.

#### 4.1.2 CULTURAL ADAPTATION

You need to clearly classify the pages on the web site under the following headings:

- Pages that require heavy cultural or regional adaptation



- Pages that require straight translation
- Pages that require no translation

Pages that require real “localization”, i.e. many adaptations to regional standards and conventions, such as marketing text, may need to be rewritten by local authors in each of the target languages. Involving in-country offices or contacts in this process is recommended. Localizing this type of content goes way beyond translation because it is an integral part of the company’s global branding initiative. Pages that contain no country-specific information, such as technical product information, can be sent directly for translation. Examples of pages that require no translation are local advertising campaigns or local office information.

**Tip:** Web sites that contain a lot of *local* content, as opposed to localized or translated information, tend to increase the comfort level of international visitors. For example, do not just translate your company information into German, but also include additional information about your German operations, your activities in Germany, or local case studies and client references.

If a web site offers online purchasing capabilities which need to be localized, ensure there is a strategy for the following:

- Local payment facilities
- Local product mix
- Local distribution system
- Local laws for taxes, tariffs, etc.

Obviously, true web site globalization in the case of e-commerce sites requires far more effort than just localizing the content.

### 4.1.3 UPDATE MANAGEMENT

It is important to consider web site localization as part of the development process, because web site content is often more frequently updated than any other medium. It is essential to use a localization vendor with strong project management skills and advanced translation technology. Develop a long-term relationship and update/maintenance plan with this vendor, as repeated and quick turnaround times will be needed from this vendor.

Essential to update management of multilingual web sites is the use of translation memory systems. For advanced, large web sites published in many different languages, the employment of a “multilingual content management system” or version control software is recommended. These systems compare new or updated documents to the pre-existing memory files, and will only extract the new and previously untranslated text. After the new text has been translated, it is automatically reinserted into the HTML pages or database and uploaded to the web site.

Several tools have been developed to monitor web site content and automatically transfer new or changed content to translators. Examples of such tools are LionTrack

by Lionbridge Technologies ([www.lionbridge.com](http://www.lionbridge.com)) and SDL WebFlow by SDL International ([www.sdlintl.com](http://www.sdlintl.com)). These tools automate the file transfer of frequently updated content and are usually integrated with a translation memory system.

#### 4.1.4 HTML PAGES

In addition to the overall setup and design of the web site, the individual HTML pages also need to be internationalized.

It is advisable to use the latest version of the HTML standard if there is a plan to publish multilingual web pages. Version 4 of the HTML standard supports Unicode and allows use of multiple languages and character sets in HTML pages. It is also possible to enable foreign language searching and sorting functions by using the right language, region, and character set encoding tags in the pages. For more information on inserting language tags, refer to the Engineering HTML section on page 264.

It is best to avoid using too many graphics with text on web pages. If graphics with text need to be used, create image files where the text is stored on a separate text layer to make them easier to localize. Keep the foreign target audiences in mind when designing web pages with images. Many countries do not yet have an advanced high-speed internet infrastructure, so restrict the number of graphics.

Where Active Server Pages (ASP) are used or HTML pages containing heavy scripting, include comments for translators, explaining which elements should be translated and where or how they will be displayed. ASP pages often contain a combination of static content, which is translatable, and dynamic information that is retrieved from a database or external files. The dynamic content should be translated in the source, i.e. the content management system or database, not on the ASP page itself.

The most efficient way to design HTML pages for localization is to create subsets for each language in which all path settings and links have already been adjusted to the target language folders and images, and where all language and character set encoding tags are set correctly. This way the localization vendor will only require this subset of the HTML files, which can be merged directly into the central web site after localization. For example, provide the vendor with the contents of a `\de` folder with HTML files that already link to the `\de\images` folder where the localized images will be stored.

Consider creating separate HTML templates or style sheets for each language or set of languages. For example, style sheets for Asian languages could define a different font set to be used on the localized HTML pages.

For more information on HTML internationalization, visit the World Wide Web Consortium web site at [www.w3c.org/international](http://www.w3c.org/international).

## 4.2 Check List

When designing a web site, follow these guidelines:

- ✓ Decide on either a central, multilingual web site or a distributed, language or region-specific web site.