



Dag Forssell was born and raised in Sweden. In 1965 he earned an M.S. Mechanical Engineering from Chalmers University of Technology, Sweden. In 1967 he and his wife Christine emigrated to the U.S. and in 1971 he received an MBA with emphasis on finance from the University of Southern California.

Dag has worked in American industry since 1967 with assignments in design engineering, manufacturing, financial reporting, marketing, and management. He is experienced as a speaker and writer on technical and leadership subjects. Three articles were published 1993-95 in *The Engineering Management Journal* and are included in his book: *Management and Leadership: Insight for Effective Practice*.

In 1994 Dag and Christine established the *Forssell Translation Team* which has successfully handled a number of translation and localization projects.

Dag and Christine can be reached at team@forsselltrans.com. Their web page is located at <http://www.forsselltrans.com>.

- [Front Page](#)
- [April '01 Issue](#)
- [January '01 Issue](#)
- [October '00 Issue](#)
- [July '00 Issue](#)
- [April '00 Issue](#)
- [January '00 Issue](#)
- [October '99 Issue](#)
- [July '99 Issue](#)



One Translator's Thoughts on Software Localization

by Dag Forssell

My wife Christine and I have participated in software localization many times. This has meant participating in bits and pieces of the process—translating strings, manuals or help files as requested by translation agents. Recently, I have been privileged to work on a large localization project in an integrated fashion, with all available resources at hand, and found it most satisfying. I don't know all there is to know about the software engineering that goes into localization, but I have experienced the process as a translator when it was organized different ways by agents and their customers. From my perspective, the way a project is organized makes a substantial difference to the ultimate quality of the project. Here is an attempt to portray two basic approaches in the form of two short scripts, one labeled *Distributed*, the other *Integrated*. Comments in the scripts reflect my experience.

Regarding program manuals, documentation and help texts, where the software has already been translated, please note *Additional information* at the end of this article.

Software localization: A translators interpretation—Script in two parts

Summary table

Script in two parts

Part I: The distributed approach

Part II: The integrated approach

Additional information

Glossary of terms

Industry standard glossaries

Informative article

Translation considerations

Summary Table

SOFTWARE LOCALIZATION

- [April '99 Issue](#)
- [January '99 Issue](#)
- [October '98 Issue](#)
- [July '98 Issue](#)
- [April '98 Issue](#)
- [January '98 Issue](#)
- [October '97 Issue](#)
- [July '97 Issue](#)

• [Thank You!](#)
by Gabe Bokor

• [Index 1997-2001](#)

Translator Profiles

• [The Making of a Translator](#)
by Louis Korda

The Profession

• [The Bottom Line](#)
by Fire Ant & Worker Bee

Legal Translation

• [La traduzione giuridica](#)
by Deirdre Exell Pirro

Arts and Entertainment

• [In the Footsteps of Giants](#)
by Robert Paquin, Ph.D.

Localization

• [One Translator's Thoughts on Localization](#)
by Dag Forssell

Translation Theory

• [Translation and Language Varieties](#)
by Magdy M. Zaky

Translators Around the World

• [The First Three Years](#)
by Timothy Howe

Translating Social Change

• [Translation as Rewriting](#)
by Berrin Aksoy, Ph.D.

Science & Technology

• [A Translator's Guide to Organic Chemical Nomenclature XXIV](#)
by Chester E. Claff, Jr., Ph.D.

	Distributed	Integrated
Salesperson involvement	Take order	Explain process
Customer involvement	Laid back	Rapid response
Translation resources	Few or none	Complete
Translator satisfaction	Minimal	Good
Translation speed, program	Normal/slow	Slow/meticulous
Translation speed, help text	Normal	Normal or a bit faster
Project Quality	Mediocre	High

Cast of characters:

Customer: Software company that needs localization (software translation)

Buyer: Customer's purchasing agent—buys translation from translation agency

Expert: Expert on customer's software—programmer or applications manager

Salesperson: Translation agent Sales representative

Manager: Translation agent Project manager

Engineer: Translation agent Technical manager

Translators: One or more

Script in Two Parts

Part I: Software localization—The distributed way: (break it down, spread it around and get it done fast)

Buyer: We want to have this program translated. Can you do that?

Salesperson: Yes, we have extensive experience with software localization.

Caught in the Web

- [Web Surfing for Fun and Profit](#)

by Cathy Flick, Ph.D.

- [Translators' On-Line Resources](#)

by Gabe Bokor

Translators' Tools

- [An Effective and Inexpensive Translation Memory Tool](#)

by Andrei Gerasimov

- [Translators' Emporium](#)

- [Letters to the Editor](#)

- [Translators' Events](#)

- [Call for Papers and Editorial Policies](#)

Buyer: Our marketing people want it yesterday. When can we have it?

Salesperson: Let's see—we are talking about 20,000 words of program translation and 200,000 words of help texts and manuals. With two or three translators doing 10,000 words a week each, translation alone will take 10-12 weeks. Let me schedule it and include time for engineering and validation by your expert. Looks like we can do it in 16-18 weeks.

Buyer: Fine, go ahead.

Salesperson: Great!! I got the order.

It appears to us that if the buyer does not understand translation, but wants the job done quickly, and the salesperson either is not familiar with translation or simply feels compelled by real or imagined competitive pressures to promise rapid delivery. Deadlines for a project are firmly established—tight deadlines that take on overriding importance.

Manager: Glad you got the order! Did you ask for reference materials? I see, they wanted delivery yesterday—I had better break this down, spread it around and get it done fast.

If buyer and sales person focus on due dates, but have little understanding of translation, they are not likely to review and discuss what reference materials exist.—Materials that translators who actually will be doing the job can use to assure consistency and quality. This means that the project manager will tell us that there are no reference materials available. We frequently ask agents to contact the customer again with a request for reference materials.

Sometimes we are asked to translate a glossary of typical terms, created by the Customer expert, as a preliminary step. Such a glossary can be used for translation of strings. Translating a glossary when context is available is fine, but translating without context is never a good idea.

Engineer: (Receives program code from customer and prepares a series of Resource Files (software strings) for translation.)

Manager: (Sends software strings (the program elements) to one translator for translation.)

The working program itself is not sent—after all, the buyer did not volunteer to provide it. Help texts and manuals are not sent to translator who will work on strings. That would just slow him or her down. Pay the standard word rate—a word is a word is a word.

In our experience, working with a large number of project managers, we find that many who have never spoken another language, or have no experience as translators, appear to think that translation is a matter of substituting a word in one language for a word in another language. If so, what the translators' backgrounds are, whether they understand the source material and how they expresses ideas in the target language does not matter.

Translator: (Translates the software strings without access to reference materials or working software.)

When we have been asked to translate software strings, we have rarely been provided a look at the help texts and manuals or the program. The program menu structure is contained in these strings and as long as the menu items are standard Windows fare, we are OK. But when you get

one or two words by themselves, unusual or strange technical or business terms with no context whatsoever, this process becomes a challenge with a most uncertain outcome.

Engineer: *(Receives the translation from the project manager and rebuilds it into working software.)*

The translation will likely require extensive revision when a linguist and the expert review it. Worse yet, it may not be thoroughly reviewed and revised.

Expert: *(Reviews program translation when program has been rebuilt.)*

Naturally, it is a good idea to have the customer expert review the program translation, and this is likely included in the project schedule. But if the expert is preoccupied with something else, the translation of help files will proceed at full speed without this review.

Manager: *(Farms out the help texts and manuals to several translators. These translators can now be provided with the program translation and asked to translate the help texts and manuals consistent with it.)*

Translators: *(Translate help text, still without access to manuals or working software, but with access to the software string translation and any other word list developed for the project as glossary.)*

If two or three translators work on the help texts and manuals, terms used in the software strings will be propagated in the help texts and manuals. This assures that mistaken translations—individual terms translated incorrectly in the software strings due to a lack of context, will be widely distributed in the help texts and manuals. Other terms that appear throughout the help texts and manuals, but not in any project-specific glossary, will likely be translated two or three different ways. This is especially true if the agent does not facilitate close communication between translators.

Engineer: *(Compiles the help files with the program and the finished software package is delivered to the Customer.)*

The end result of this process is software localization of questionable quality.

Chorus: Let's all hope the Customer's end users don't complain.

Part II: Software localization— The integrated way: (Get it together, keep it together and get it done well)

Buyer: We want to have this program translated. Can you do that?

Salesperson: Yes, we have extensive experience with software localization. We have found that it is best to work closely with your company's programmers and application managers. Will they be available to work with us?

Buyer: Yes, I think so. Our marketing people want it yesterday. When can we have it?

Salesperson: Before we discuss delivery, let me ask you what your marketing people expect in terms of quality. Does your company want a quick and dirty localization, or do you want something your end users will be happy with? The difference may be a few weeks.

Buyer: Why do you ask? We want quality, of course!!!! When can we have it? If you can't deliver yesterday, I will find someone who can!

Salesperson: Let's see—we are talking about 20,000 words of program strings and 200,000 words of help texts and manuals. We will find a translator who is familiar with this kind of application to work on the strings and provide her with all the help text and working software as reference. You do have a working copy of the software I can take with me, right? Also, we will need all manuals that go with the application so we can pass them on to the translator. Anything that will help the translator become familiar with the application, menu structure, dialog boxes, error messages etc. The translator should have plenty of time with the strings so she can become familiar with the terminology you use. Who is your expert on this application? We will need to stay in close touch because, in our experience, the translator will have lots of questions about idiosyncratic terms, abbreviations, confusion and inconsistencies she discovers in the source language.

Just doing a thorough job with the program strings as a first step and preparing a glossary at the same time will take about four weeks. After that, two or three translators can use the glossary the first translator has created to keep the terminology consistent throughout the help texts and manuals. That phase will take about 10 weeks. Let me plan it out with time for engineering and validation by your expert. Looks like we can do it in 18-20 weeks. With an up-front emphasis on strings and a glossary translated in context, this is just two weeks more than a somewhat faster parallel effort that leads to much lower quality.

Buyer: Sounds reasonable to me. Let me get you a copy of the program right now, check availability of support materials, and confirm that our expert will be available.... Fine, go ahead!

Salesperson: Great!! I got the order.

Manager: Glad you got the order! You sure covered all the bases and got what our translators will need. You got it all together. I will keep it together and get it done well.

Engineer: *(Receives program code from customer and prepares a series of Resource Files (software strings) for translation.)*

Manager: *(Sends software strings, all help text and a working copy of the program—all together—to a translator who understands the subject matter and is familiar with software and operating system practice (e.g. Windows and Microsoft glossaries).)*

Translator: *(The translator can now review all materials thoroughly, translate the software and develop a glossary. The customer's expert is available to answer questions.)*

My recent experience, working with an experienced, cooperative manager and a responsive expert, has been that I can translate the strings (the program elements—menu structure, dialog boxes and messages) and relate these elements to the way they are explained in the help texts and manuals. I can review the program and see how the menu structure is set up—what the functional relationships are between various text elements that are extremely brief and, taken alone, cryptic.

I have found that having access to this context makes a major difference. I have been able to translate a great number of terms, whether business, technical or computer related, not based on what the dictionary says or how these terms are ordinarily used, but based on how they are used or misused in this particular program—based on this particular expert's or his company's individual, idiosyncratic lingo.

Rather than making fast and loose translations of individual terms out of context, I am able to look them up in the help text, read what they mean and see how they are used. I am able to review the menu structure and the dialog boxes in the working software. Naturally, this takes time. In fact, I found this translation work far more meticulous and time-consuming than anything I have experienced before. But the work is satisfying because I know I can do a good job. I am confident that this localized program will make sense to its user.

While I look up terms in the help texts and manuals and decide how to translate them, I take notes for use later on, when I and my team translate the help files. If a term exists in abbreviated form in the software strings, I take note of it and its translation in its full form as well, because that is how it appears in the help texts and manuals. This, together with the strings themselves, becomes the basic glossary for this project.

Engineer: *(Receives the translation from the project manager and rebuilds it into working software.)*

The program will require review and abbreviation of some terms to fit available space in dialog boxes, but revisions will be minimal when a linguist and the expert review it.

Expert: *(Reviews program translation when program has been rebuilt.)*

As with distributed translation, it is a good idea to have the customer expert review the program translation, and this should be included in the project schedule. But if the customer expert is preoccupied with something else, the translation of help files will proceed at full speed without this review. The difference with integrated translation is that even without the review, I am confident that the program and the help texts and manuals will hang together.

Manager: *(Assigns the help texts and manuals to several translators.)*

Translators: *(Translate help text with access to the glossary inherent in the program translation as well as additional glossary notes made during the string translation, and with access to manuals and working software.)*

Terms that appear throughout the help texts and manuals can now be translated consistently. Additional coordination between several translators can enhance consistency even more.

Engineer: *(Compiles the help files with the program and the finished software package is delivered to the Customer.)*

The end result of this process is software localization of high quality.

Chorus: Let's all take pride in a job well done.

Additional information

Glossary of terms:

Program, Resource files, .RC files, Strings: These terms refer to the visible part of the software, the user interface: menus, dialog boxes, error messages—any text that appears in the program and its operation and

needs to be translated.

Help text, manual, documentation: All the other text that relates to the program interface, describing it, explaining what happens, troubleshooting tips, installation instructions etc. Needless to say, all this should be compatible with the terminology used in the program interface itself.

Industry standard glossaries:

Microsoft glossaries: Microsoft publishes glossaries based on the .RC files for its operating systems and most of its programs. See: <ftp://ftp.microsoft.com/developr/MSDN/NewUp/glossary/>.

Apple glossaries: The last time I found any Apple glossaries was back in 1996 for OS 8.6. Apparently Apple no longer makes glossaries available; however, I recently found an English-only description of Mac terms as a pdf file at <http://developer.apple.com/techpubs/macos8/Glossary/IMGlossary.pdf>, still for OS 8.

Informative article:

Facets of Software Localization, A Translator's View by Per N. Dohler is an excellent, comprehensive article published in an earlier issue of the *Translation Journal* at <http://accurapid.com/journal/softloc.htm>, which I found when I was looking for information on Resource files.

Translation considerations:

Pricing translation: It should be apparent from my discussion in the scripts in this article that a standard word rate for translation of normal text should not apply to translation of .RC program string files.

Software already translated: If the program has already been translated and the customer wants some documentation translated, the **translator should receive the .RC files** in both the source and target languages, or some combination of these, such as segmented Trados files. These files provide complete insight into the terminology and all messages tucked away in the program. They are the key to good, consistent translation of accessory documentation. As the translator works on the documentation, the translator should be welcome to ask about and/or change mistakes discovered in the software strings.

The program itself should also be provided—in both languages. The program provides a look and feel for the program that .RC files do not. But programs do not easily surrender equivalent text and messages. Programs or demos are not equivalent to the Resource files and vice versa.

When a customer solicits a bid: If your agent representative is not intimately familiar with localization, please don't submit a bid based on a casual word count. Review the assignment carefully and consult an experienced translator who is familiar with the program's subject matter as well as localization.

© Copyright *Translation Journal* and the Author 2001

Send your comments to the [Webmaster](#)

URL: <http://accurapid.com/journal/17softloc.htm>

Last updated on: 05/03/2003 22:05:37