

Volume 4, No. 1
January 2000



Tibor Környei, 46, was born in Budapest, Hungary. After graduating from the technical university, he worked as a civil engineer for ten years. In the meantime, he learned English and acquired a translator/interpreter diploma at a postgraduate course. In 1990 he changed professions and began to work as a freelance English-Hungarian translator. In addition to doing translation jobs in his field of specialization, he translated several novels and economic-financial textbooks for publishers. Together with an American translator, he translated various materials for the Foreign Ministry for two years. In recent years he has participated in a number of software localization projects.

Founder and moderator of the Translators' Electronic Forum in Hungary (since 1995). Co-editor of a Newsletter published by the Association of Hungarian Translation Companies. As a guest lecturer, teaches Computer-Aided Translation at a postgraduate translator training course at ELTE University, Budapest.

Tibor Környei can be reached at:
tkornyei@wordfisher.com.

- [Front Page](#)
- [October '99 Issue](#)
- [July '99 Issue](#)



WordFisher for MS Word

An alternative to translation memory programs for freelance translators?

by Tibor Környei

Computer programs making use of translation memory (TM) have been mentioned in several articles published in the *Translation Journal*. These programs were originally developed for the software localization industry to support the translation of high-volume texts that contain a large number of repetitions, are frequently updated, and require quick turnaround times. The use of TM also represents invaluable help in translating technical documentation in the automotive and aeronautical industries, which often use TM projects developed in-house for certain large projects. Most commercial TM programs of today were conceived in the course of such projects.

Reasons for the TM buying frenzy

In late 1999, we were witnessing widespread popularity of TM programs. This phenomenon can be attributed to several reasons:

- Over the years, translation bureaus and translators were led to believe that TM programs represent the cutting-edge technology in the translation industry. The fact that these programs are actually special tools to support special projects somehow got lost in the din of self-promotion. Translators who believed they could effectively work without such aids were suddenly made to feel insecure.
- More and more translation bureaus have looked for translators who used TM programs. The intent behind this has, unfortunately, rarely been to improve quality, but rather to reduce costs.
- Price competition among TM programs toward the end of 1999 has made these programs, which used to be

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

● [Happy 2000!](#)
by Gabe Bokor

● [Index 1997-2000](#)

Translator Profiles

● [Love, Languages, and Translation](#)
by Peter Griffin

The Profession

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

● [When Trust Is Broken](#)

● [ASTM Standard for Language Translation](#)
by Steve Lank

● [Translating the Web: Into the Future](#)
by Jan Oldenburg

● [Science or Translation?](#)
by Maria Karra

Translators in the Media

● [Translators and the Media—Part 1](#)

● [Translators and the Media—Part 2](#)

Business & Finance

● [Translating a Brazilian Balance Sheet](#)
by Danilo Nogueira

extremely expensive, affordable.

- Instead of modules, complete packages were being offered (Trados, Transit), at an attractive price, which made translators believe that they could purchase an essential tool with a one-time investment.

WordFisher cannot be considered an alternative to TM programs where these are used for their originally intended purpose. But it can be an alternative wherever the nature of the task does not justify the use of a heavy-duty TM program.

According to the information found on Trados's home page, around 4000 copies of the *Freelance* package were sold worldwide during the few months preceding early December within the company's 15th Anniversary Special Offer.

Does the nature of translation projects justify this volume? Are there this many large projects in the translation market? I am not even sure that many of the truly large projects ever get to the free translation market. If a large company has an ongoing

need for translation, is it not more economical for it to purchase or develop the program that best satisfies its specific needs and perform the work in-house or by building up a relationship with a bureau that specializes in that industry?

If the amount and nature of projects do not justify it, what explains this buying frenzy? On the one hand, it is actually due to panic, i.e., fear of technical obsolescence and loss of orders. On the other hand, it is because these programs offer a number of functions that are actually helpful in the translator's work. I would emphasize two factors: *the emergence of corpuses* and *automatic performance of repetitive tasks*.

Dictionaries v. corpuses. Dictionaries are the traditional aids to translators. With the spread of computers, electronic dictionaries and glossaries have also emerged. These were initially "copies" of their paper-based counterparts, but had much better search capabilities due to the nature of the technology. With the increase in computer power and the popularity of CD-ROMs, encyclopedias have also become very popular with translators. Encyclopedias specifically designed for computers, with no paper counterparts (such as Microsoft Encarta) were born. Multilingual corpuses (such as the Bible and legal corpuses) also emerged.

Automatic replacement of repetitive texts. Depending on their type, translation projects contain a certain number of repetitive segments. These segments may be words, specialized terms, standard phrases, or even complete paragraphs. While most search and replace operations can be performed with the built-in functions of word processors, TM software promises—and increasingly also offers, although not to extent as many believe prior to purchasing—"intelligent" and "automatic" search and replace over the entire project.

Translating Development

● [Neologisms in International Development](#)

by Alexandra Russell-Bitting

Arabic

● [The Arabic Language and Folk Literature](#)

by Srpkó Lestarić

Science & Technology

● [A Translator's Guide to Organic Chemical Nomenclature XVIII](#)

by Chester E. Claff, Jr., Ph.D.

Caught in the Web

● [Web Surfing for Fun and Profit](#)

by Cathy Flick, Ph.D.

● [Translators' On-Line Resources](#)

by Gabe Bokor

Translators' Tools

● [WordFisher for MS Word](#)

by Tibor Környei

● [Translators' Emporium](#)

● [Translators' Events](#)

● [Letters to the Editor](#)

● [Call for Papers](#)

Since no other program offered translators a solution for building and managing corpuses or an intelligent project-level search and replace function, it is understandable that many people considered the purchase of a TM program worthwhile from purely rational considerations.

Many buyers are or will be disappointed. I know several translators who hardly use the program they purchased or do not use it at all. Tunneling through hundreds of pages of a manual, learning the use of a new interface, the concept of databases, etc. are particularly difficult for those who would like to focus on translation, rather than technology.

Could it be simpler?

I've been developing the *WordFisher for MS Word* macro set since 1994. It started out as a glossary management program, but later, as I became familiar with TM programs, I became "envious" of their functions and attempted to implement as many of them as possible in the macro language of *Word*. I realized that there is a gap between advanced word processors and heavy-duty TM programs. *WordFisher* intends to fill this gap, and its current version can be regarded as a full-fledged translation aid program.

The new version of *WordFisher* requires *MS Word 6.0* or above, but it also works under *Word 2000*. The program is written in the *WordBasic* language. For the translator, it resembles a TM program, but provides a simpler interface in *Word*. The main functions of *WordFisher* are:

- Handling (multi-file) translation projects, automatic indexing to help navigation within the files in the project, and integrity control (existence of files, integrity of indices);
- Searching in the project, context check (this function is usually neglected by TM programs; they normally search in previous translations only, although context within the text to be translated is often more useful);
- Replacing in the project, even multiple replace on the basis of a list (in practice this means the possibility of pre-translation based on a list of expressions);
- Automatic glossary preparation during translation (logging the replacements);
- Automatic building of a bilingual corpus (prepares a table from the source- and target-language sentence pairs);
- Checking the consistency and completeness of the finished translation (on the basis of the sentence pair tables).

When translating by overwriting, the translation's bilingual corpus is built automatically. Corpuses can be later input in any TM program without time-consuming preparatory work. This function alone makes working in *WordFisher* worthwhile. *WordFisher* has a built-in aligner to convert previously done translations into corpuses.

Although corpuses can be searched by *WordFisher*, larger corpuses are better handled using external programs. In my experience, *dtSearch* (<http://www.dtsearch.com>) is ideal for this purpose. *dtSearch* uses indexing, can process almost any file format, and can even search using fuzzy technique if needed. The result of a search can be better viewed in *dtSearch* than in most TM programs.

Organizing a translation project with the help of WordFisher

It has been shown under actual working conditions that projects theoretically (by the agency's theory) requiring a TM program can be handled using *WordFisher* and *dtSearch* without affecting quality or speed.

All TM programs are capable of exporting their translation memories into a parallel text and of importing parallel text pairs into their translation memories. These functions are used in fitting *WordFisher* to TM programs.

The customer sends the exported translation memory and the glossary of the project. If their combined size does not exceed 1 or 2 MB, *WordFisher* can handle the reference material by itself. If there is more reference material, it is convenient to combine it with the files received and to handle them together, for example, using *dtSearch*.

The translation project is created in *WordFisher* from the files to be translated. This process is automated—no user intervention is required. Translation is then performed by overwriting. During the translation process, the reference material must be constantly monitored, and whatever is found there must be taken into account. Repeating expressions only have to be entered once; *WordFisher* does the replace in all the files.

Upon completion of the translation, the program prepares the tables containing the source-language and target-language sentences. These can be returned to the customer along with the translated files, and the customer can then enter them into the translation memory of his own TM program.

Finally, an answer to the question

The answer to the question posed in the title of this article: *WordFisher* cannot be considered an alternative to TM programs where these are used for their originally intended purpose. But it can be an alternative wherever the nature of the task does not justify the use of a heavy-duty TM program, i.e., in smaller *MS Word* projects, which constitute the bulk of most translators' everyday work.

The home page of *WordFisher* can be found at:
<http://www.wordfisher.com>

(c) Copyright *Translation Journal* and the Author 1999

Send your comments to the [Webmaster](#)

URL: <http://accurapid.com/journal/11wf.htm>

Last updated: 12/01/2003