

Background

In the last 15 years we have used various search engines. Occasionally we had to switch because of technical improvements in the world (e.g. 16-bit to 32-bit, hard disk vs. CD-ROM, local vs. network, and access via web technology).

In 2002 we received a request from one of our customers to enhance an already running Search & Retrieval application with a very specific method of proximity searching (area search). The major requirements were:

- similar behaviour/functionality/performance compared to earlier search engine(s);
- boolean operator support (and, or, not);
- wildcard support for "*" (zero or more characters);
- wildcard support for "?" (exactly one character);
- proximity support with variable distance (e.g. bike near50 car);
- able to search collections of PDF files (Portable Document Format);
- able to highlight hits within the native PDF viewer (Acrobat Reader);
- able to be used on CD-ROMs and/or DVDs;
- able to be used in ASP-environment on Windows IIS 5.x and higher (intended use: Intranet);
- minimal dependencies to other modules, libraries etc. to prevent difficulties in setup/installation.

Investigation

In 2003, comparing several commercial packages, we realised that it was difficult to meet all the above requirements. In order to increase flexibility in the future we started looking for "open-source" search engines and we soon discovered the vast range of options and flexibility of Swish-E. Particularly useful was the multi-platform characteristic (including Windows, our commonly used platform). Although Swish-e did not (yet) meet all our requirements, we saw possibilities to enhance Swish-E, so in the end it would probably meet all our requirements. We started investigating the source code, cleaned up many files for incorrect CR and LF characters, and finally we were able to build the multi-threaded search library on a Windows platform (using Microsoft Visual C++ 6.0). We also succeeded to statically link the multi-threaded search library into the SwishCtl library (COM-technology), so it could be more easily used in many different environments on a Windows platform (JavaScript, ASP with VBScript, Win32-executable).

Many of the requirements were already met by the initial build of Swish-E v2.3.x (mid 2003). In November 2003, we upgraded to the official/stable version at that particular moment: v2.4.0 of October 27, 2003. At the same time we were working on a solution to decouple hit-highlighting in PDF from the actual search engine used. This would allow us to easily switch to another search-engine if required in the future. We managed to create a text-extraction tool for PDF-files, which produces a text-file to be indexed with Swish-E and a separate-file, which can be used for highlighting. This approach enabled us to minimize the required modifications to Swish-E, so it should be relatively easy to upgrade to a future version of Swish-E (e.g. to the stable version at this moment v2.4.3).

The "only" required modifications to Swish-E, still left were:

1. enhanced wildcard support (also question mark "?" support to indicate a single character; e.g. "PUBL?C LICE??*");
2. proximity search support by means of "nearx" operator (e.g. near50), which is actually a special case of the Boolean AND operator;
3. ultimate filenames should follow the 8.3 rule (docx.idx and docs.prp for ISO 9660);
4. replace message boxes in SwishCtl with normal error codes;
5. change Init call in SwishCtl to prevent registry access;
6. remove DLL dependency of zlib.dll and atl.dll.

Realisation

Although the actual changes to the official Swish-E source code were relative minimal, it took a few months to analyse, debug, enhance, debug again and test the final result in many Windows environments. The modifications to Swish-E search library are not Windows specific and could easily be incorporated in the official Swish-E source (if desired). The modifications to SwishCtl are only relevant in a Windows environment, since the control itself is already ATL/COM-based.

Item 1 (wildcard): derived from the already present support for wildcard "**";

Item 2 (proximity search): derived from the already present operator "AND". The position of words are compared and if they are in the specified range, the item found is included in the final result, otherwise it is discarded. Although the initial implementation was a simple proximity search, we had to enhance it to make it more "area" aware. The example "car near50 bike near50 vehicle" now only returns the item if all distances are within 50 words from each other, so "car" should be no further away than 50 words from "bike" **and** "car" should be no further away than 50 words from "vehicle". If in this case "car" was 50 words from "bike", "bike" was 50 words from "vehicle", but "car" was 100 words from the nearest "vehicle", the item would **not** be included in the result.

Item 3 (filenames): some string manipulations to make sure ".prp" is used.

Item 4 (message boxes): added an extra error code, which is set whenever a message box would normally show; at the end this specific error code and the default Swish error code are checked; if one of them is set, the error code is returned to the caller.

Item 5 (Init call): added specific string to parameter of Init call; if that string is detected (and also "+"), then the specified path is used to locate the index file, instead of the indirect way via the registry; if our specific string is not part of the parameter, SwishCtl works just as before, so it is downward compatible.

Item 6 (dependencies): "zlib" is built as a static library and subsequently linked statically into SwishCtl; for "atl.dll" a special flag is added at built time so the so-called registration code is incorporated in SwishCtl.dll itself, this implies that one does not need to use the separate "atl.dll" anymore.

Result

The final result is SwishCtl.dll (generated for Windows and 276KB in size), which can be used in a standalone environment (CD/DVD) and/or in a web environment (IIS). No other files have to be distributed, as far as just searching is concerned. Other functionality (e.g. PDF hit-highlighting), is located in a few other libraries and is fully decoupled from the used search engine. This provides a highly flexible and scalable solution.

Conclusion

Swish-E proved to be a fast/ flexible open source search engine, which can even be further enhanced, because it has been written in "relatively easy" portable C. Other advantages are the small footprint and it has proven to be a well written, multi-threaded building block, which runs efficiently and is very reliable. It also works in an ASP-environment on version IIS 4.0 and higher (already for the last couple of years).

Although not required in our current use of Swish-E, some known shortcomings of Swish-E are:

- no multibyte support (already on the wish list of Swish-e);
- relative poor performance when using wildcard with only 1 or 2 preceding characters (e.g. 3* or 31*); in this case it makes a huge difference whether the index files are present on the hard disk or CD/DVD.

Downloads (Windows binaries)

1. [swish-e.exe](#) to build Swish-e index-files (.idx and .prp);
2. [swishctl.dll](#) multi-threaded ATL/COM component for searching;
3. [libswish-e-mt.lib](#) multi-threaded library for searching (static linking).

Source code (copy these files on top of "Swish-e version 2.4.0, October 27, 2003")

1. [src-swish-e](#) modified sources containing earlier mentioned enhancements;
2. [src-swishctl](#) full source code of slightly modified swishctl component.

Full documentation and background of Swish-e can be found at <http://www.swish-e.org>.