

OttoSource32 - A File Browser / Stream Editor



quaerendo invenietis.

Purpose

OttoSource lets you search for text in files and browse them on the screen. With character-formatted files you can also make changes using various stream editing features. It includes support for very large files and the ability to search and browse non-text items.

Prerequisites:

Windows 95 or other WIN32 system.

Constraints:

The following limits are imposed:

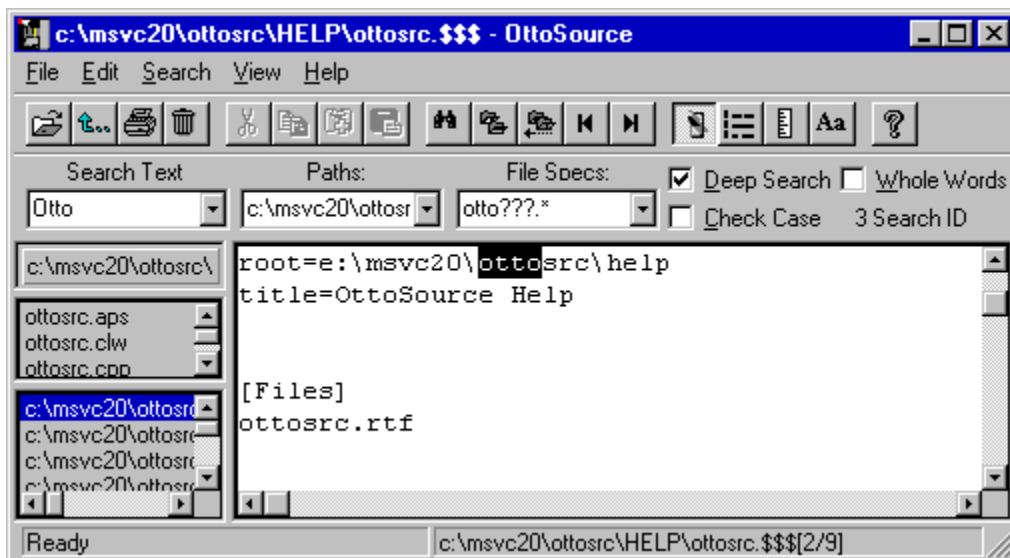
file size: available memory
record number: max(LONG)
text record size: 4k

Installation:

☞ Copy OTTOSRC.EXE, and the help file OTTOSRC.HLP to any suitable directory.

The OttoSource Display

OttoSource presents a single window with several configurable panes. At the top are (optionally) a toolbar and options bar. Below are three splitter windows on the left and a file viewer window on the right. When you finally select a file, it will appear in the file view window. Before you can get anything out of OttoSource, you have to establish a few parameters. The OttoSource screen looks like this:



Choose **View/Toolbar** to display the toolbar, and **View/Options** to show the option bar.

You can adjust the splitter windows by moving the mouse over one of the heavy bars and dragging it to resize the pane.

On the left are 3 splitter panes that display the current status:

Location	Name	Content
top	Directory Status	lists current directory
middle	Selected Files	files that match the file extension spec & path
bottom	Searched Files	results of most recent file search

Choose **View/Ruler** to display a floating character ruler with 80 positions.

Toolbar

OttoSource displays an optional toolbar that gives access to many of the menu options.



From left to right, the options are:

- *Open file
- *Change directory
- *Print file
- *Delete file

- *Cut to clipboard
- *Copy to clipboard
- *Append to clipboard
- *Paste from clipboard

- *Find text
- *Search files
- *Reload from file history list
- *View Prior Search Results
- *View Next Search Results

- *Toggle selection bar on/off
- *Toggle line numbers on/off
- *Display a floating ruler
- *Select font

- *Help

Unavailable options are dimmed and inactive.

Ruler

Click the Ruler button or choose **View/Ruler**. A floating ruler which numbers the character positions 1 - 80 is displayed. You can drag the ruler around and line it up as needed. The ruler uses the same font as the file view window. The ruler is only useful if you choose a monospace font such as Courier where each character has the same width. You can only display one ruler at a time. To quickly close the ruler click on the title bar with the right mouse button.

Using OttoSource

Briefly, to use OttoSource, you need to enter *file selection* criteria, optionally *search the disk*, and *choose a file* from one of the result windows. The file is then loaded into the file viewer on the right.

Alternately you can choose **File/Open** to select a specific file to view or specify a filename on the command line. You can edit, copy text to the clipboard, and save the current file to a new name.

Selecting Files by Filename & Directory

To start with, you must indicate the kind of files (extensions) that OttoSource should look at and what directories to look in (paths). We will call this **selecting** files by name - the results are displayed in the *Selected Files* window.

To make a selection, be sure the toolbar and option bar are visible. There are two fields to fill out:

Paths: a comma separated list of paths to search [optional]
File Specs: select filenames using these patterns (standard DOS wildcards apply)

To do anything useful with OttoSource32, you must have at least one entry in the File Specs field. The default is "*. *".

For example, to search for all Windows write file that begin with the letters PAC, enter "PAC*.WRI".

The Paths field defaults to the current file directory. OttoSource32 uses these two criteria to build the list of files in the Selected Files window.

Each "field" is actually a combo box that remembers the entries you type (they are also saved from session to session). You can recall a prior entry by pulling down the combo list box and selecting an item. This way you will build up a list of your favorite search directories and file types which you can use to quickly set things up.

To change the current directory, click the 'Directory' [...] button on the toolbar, choose **File/Change Directory** from the menu, or right click on the directory display pane. Use the dialog box to navigate the disk, then click OK. The *Selected Files* window updates automatically.

Alternately choose **File/Open** and open a text file directly.

Selecting Files by Content

You can further refine your set of files by specifying a text string in the "Search Text" field and clicking the **File Search** toolbar button/menu item. OttoSource will open each file in the *Selected File* list and look for the Search Text that you have specified. When it finds a match, the file is added to the *Searched File* window.

If you check the **Deep Search** option, OttoSource checks files in each specified directory (in the Paths field) and *all* subdirectories, otherwise it sticks with the specified ones. If you choose **Check Case** then the case must match the target string exactly. If you choose **Whole Word** than only whole words that contain the search string will match.

Please note that the *Searched Files* window does not just reflect files in the current directory - it lists everything found in the most recent search.

If you specify "\?" (w/o quotes) as the search text, OttoSource will list all the file names that match the file pattern and drive/directory regardless of the file contents. This is an easy way to search the entire disk for a set of file names that match a certain criteria.

You can print a list of the hits by choosing **File/Print Searchlist...** OttoSource does two things. First, a text list of the files is copied to the clipboard. Second, the list is placed in the File Viewer window and a Print Preview is initiated. You can inspect the output, continue with the print job or cancel.

You load a file into the **File Viewer** by clicking a file name in either the *Selected Files* window or the *Search Files* window. You only need to click once.

Searching Non-Text Files

OttoSource will read and search both text and non-text files. When searching non-text files, all control characters (bytes with an ASCII value less than space), are converted to spaces.

Proximity Searches

You can specify two search strings to match as follows. Enter both the strings in the 'SearchText' field separated by a '[nn]' entry which specifies how close together the strings must be found. This 'proximity' field can run from 1 to 80 characters. For example, to search for the strings 'Elizabeth' and 'Overmyer' from 0 to 5 characters apart, enter:

```
Elizabeth[5]Overmyer
```

in the 'Search Text' box. The number in the brackets is inclusive - it matches if the two strings are 5 characters apart or less.

You can include the search string '[n]' as a regular string by preceding it with a '\' escape character. For example,

```
Elizabeth\[5]Overmyer
```

searches for the simple string 'Elizabeth[5]Overmyer' in the source file.

hints -

Spaces are important in the search string - they count as just another character to match. Also, a proximity search proceeds by first locating the first string (to the left of the brackets). If you enter a string with very few characters, there will be a lot of matches, and the search will take a long time to complete.

The File Viewer

The file view window lets you browse the file you've chosen above. You can click the Find Text button on the toolbar or choose an **Edit/Find** option from the menus to search for text. OttoSource will hunt up the next line that matches the "Text Search" field. If found, the window scrolls to the matching line and highlights the text. The Check Case and Whole Word options apply here too.

Various keys are available for moving around in the file:

Home / End	move to top and bottom of the file
Pg Up / Pg Dn	move up and down by a screen full of lines
Up / Down ↑ ↓	move one line at a time.

The scroll bars also work.

You can select text by clicking on a line, and dragging the cursor up or down - the lines are redrawn in reverse video. If you go above or below the fileview window, the text automatically scrolls into view and extends the selection. To end your selection with the mouse, move the cursor back into the file view window, and release the mouse button. If you release outside of the window, the selection is cancelled.

Alternately, you can use the keyboard. Start a selection by pressing the Space bar - the top visible line is selected. To extend the selection, hold down the shift key, and use the Home/End or Up/Down arrow or Pg Up/Pg Dn keys to extend the selection. The Esc key cancels any existing selection.

Once you've selected some text, you can copy it to the clipboard by choosing **Edit/Copy** from the menu. If desired, you can append it to existing text on the clipboard by choosing **Edit/Append**.

You can display line numbers at the left of the file view window by choosing **View/LineNumbers** or clicking the Line Number icon.

If you double-click on a word, it is copied to the Text Search field.

Previous File/History List

Choose **Search/History** to select a file name from a list of the 25 most recent files displayed in the file view window. There is also a toolbar button to activate the History window.

Non-Text Files

OttoSource will read any disk file and display the characters it contains as a series of 80 character records. Any control characters (characters less than 'space') are converted to spaces. This lets you search and browse formatted word processing files, spreadsheets, even .exe files for any text they contain. There is additional overhead in displaying non-text files - that is, they take longer to read and display. But, there are fascinating things to be found in binary files, *quaerendo invenietis*.

When you do a **find** operation on a non-text file, if the match spans two display lines, OttoSource correctly scrolls to the first line and tries to highlight the string. Basically it uses a less exact method for the highlight so it may get confused or highlight nothing at all - then it's up to you to find the string.

Pasting Directly from Clipboard

If the File Viewer window is empty, you can create a new document from text on the clipboard by choosing **Edit/Paste...** Whatever text is on the clipboard will be loaded as a new text document into the viewer window. Pasting is allowed only for 'text' files. If you have a non-text file and must paste into it, you can save a text version of the file using **File/Save As**. Then open this text version, and all the editing features are available.

Editing Files

The **Edit** menu offers several options for altering records. There is a basic search and replace option, as well as several ways to change and delete whole records. However, OttoSource is stream-oriented. What you do not have is an interactive editing environment that lets you edit/change/and insert text directly on a line. There are many other editors that offer those options.

Only a few editing options are available for **non-text** files. You can copy highlighted lines to the clipboard and run **Find/Find Next** operations. If you have a non-text file and must edit it, you can save a text version of the file using **File/Save As**. Then open this text version, and all the editing features are available.

All changes are made to the **in-memory** image of the file - NOTHING is written to disk until you choose **File/Save As...** This applies to all the OttoSource editing options.

Find/Replace

Choose **Edit/Find Replace...** to bring up the standard Windows Search and Replace dialog. This dialog lets you enter a search string and a change-to string, find the next occurrence, change it, or globally find and change all such strings.

One peculiarity of OttoSource - it makes only one change per line for each pass of the file. When it has read the last line, it sees if there are any more strings to change, and asks you if you wish to make another pass. When OttoSource finds a hit, it scrolls the window to the matching string and highlights it (unless you have chosen Change All which suppresses the display until all changes have been made).

Stream Editing Features

The stream editing features of OttoSource are useful for globally manipulating the contents of a text file. Your commands select a subset of records from the file and change them in some predictable fashion. Most stream-editing functions are limited to text-only files - when unavailable, a menu option is grayed. There are several stream editing functions:

Edit/Delete

Deletes the currently selected (highlighted) records.

Edit/Paste

Pastes the contents of the clipboard (text only) below the currently selected line.

Edit/Set Record Length...

Pads or truncates the length of each record (as needed) to the number of characters you specify. Records are padded with spaces. When you specify the record length, you do not count the terminating CR/LF characters - OttoSource adds them automatically.

This feature is especially useful in shaping up text files for import to database programs which expect fixed length records, or creating files to upload to mainframes.

Edit/Find Delete...

Finds and deletes records that match certain criteria which you specify. Your options are presented in a dialog box as follows:



The Record Selection criteria determine which records from the file are considered a 'match', while the OnMatch Option specifies what happens to the records. The MatchWhat string is matched against each record in the file- if a **Starting-at** is given, then match begins only with the column specified. Match Case and Whole Word are applied if checked. If Starting-at = 0, then the match is applied to the whole record. Once a record is identified as either a Match or NoMatch, the OnMatch Option is applied and the next record is read. This continues until the end of the file.

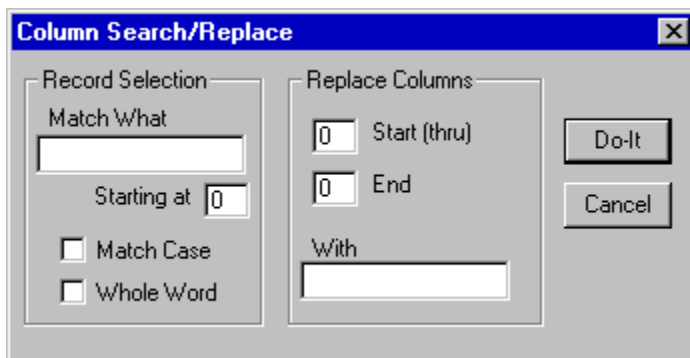
Available for text-only files.

This option is useful for quickly subsetting a file if the offending records are uniformly identifiable.

Note that the records are deleted only from the memory image of the file - there is no change to the disk until you use **File/Save As...** to actually write the file to disk.

Find/Replace Column...

Lets you edit the file stream based on matching strings in certain column positions. You fill in the following dialog box to specify the action:



The underlying logic is to select a group of records based on the Record Selection criteria, delete the characters from Replace Column Start through End, and insert the With string. The whole action is completed before updating the screen. Here are the details.

- * Records are selected based on the MatchWhat string (and MatchCase/WholeWord options) at the Starting-at column. If MatchWhat is empty, *all* records are selected. All the characters in the MatchWhat string are compared to the record.
- * You must always supply Start and End values for **Replace Columns**. Start must be > 0, End can be >= 0.
- * If **Replace Columns With** is empty, columns from Start through End are simply deleted in all selected records.
- * If **Replace Columns End** is less than **Start**, the **With** string is inserted at the Start position

and the source string is appended to itself beginning at **End+1**. This lets you simply insert strings, or even duplicate column positions.

Examples:

A record with 9 characters looks as follows:

```
123456789 <column position>
AdougBCDE <record>
```

Here are 5 sample criteria and the resulting changes:

What	Start	Start	End	With	Record
oug	3	2	4	aaabc	AaaabcgBCDE
< >		2	1	XXX	AXXXdougBCDE
< >		1	5	FOO	FOOBCDE
< >		7	8	< >	AdougBE
< >		1	0	YYY	YYAdougBCDE

Saving Changes

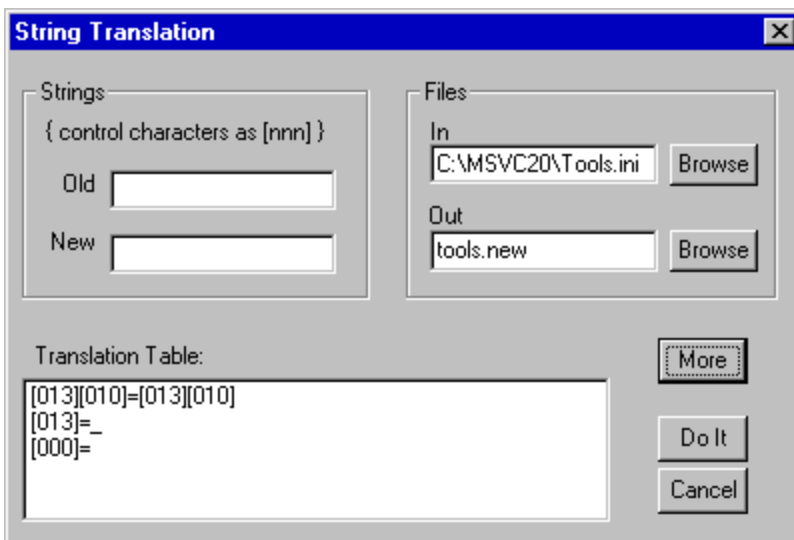
All edits and deletions are applied to the in-memory file image. Your changes are only written to disk when you choose **File/Save As...** If you choose to overwrite an existing file you will be prompted first.

Translating Files

As a stand-alone feature, OttoSource supports character translation for file streams. The idea is that you specify strings of characters that you wish to translate in one file when it is rewritten to a second file. This feature is similar to search and replace except that it operates on **all** kinds of files, not just text, and supports all valid character codes (0 - 255) including control codes. The file translation option does not load the target file into the file window, since that operation suppresses control characters which are the *raison d'etre* for this option.

Beware that you can destroy the logical consistency of your target file if you exercise this feature carelessly. To help reduce the risk, you are prevented from overwriting the input file.

To begin a file translation, choose **File/Translate...** and fill in the dialog box. There are several things you need to specify:



Files

Specify the **In** file and **Out** file. You can either type in a complete path name or use the **Browse** buttons to locate an existing file. The **In** and **Out** names must be different.

Strings

The **Old** and **New** edit boxes let you specify a pair of translation strings. The strings consist of characters that you type or specify. You can enter any keyboard character by just typing the character(s) in the edit box.. To specify a control character, make a special entry in this form:

[nnn]

where nnn = ASCII or ANSI decimal code of the character you wish to specify. Be sure to include the brackets and 3 numeric characters. You can intermix regular characters and control characters to form your string. Each string can be up to 999 bytes long, and you can have a large number of translation pairs - more than you could stand to type. Here are some sample strings:

String	Meaning
abc[097][098][099]	'abcabc' - two ways to say the same thing
country[013][010]	'country' followed by CR/LF characters
[009]Now	'Tab' followed by string 'Now'

Translation Table

Once you have specified a translation pair, click **More** to move the strings to the Translation Table. The table is just for reference but you will find it handy if you are specifying several translations. The old and new strings are separated by an '=' sign which is inserted for legibility.

When you have added all the strings you wish to translate (and specified the **In** and **Out** files), choose **DoIt** or **Cancel**. If all goes according to plan, your **In** file is opened and copied to a block of global memory, matched byte by byte against the translation strings, blocked and rewritten with changes to the **Out** file. This process requires that you have enough free memory to hold the entire byte stream.

In general, the order of the translation strings is immaterial, with this exception. The comparisons are made in the order you entered the strings, and stop with the first hit. If a match is found at a given offset, the bytes are translated and moved to the output file. No further attempt is made to match at this point in the file - instead the file pointer is bumped up to the next unmatched byte. This features lets you 'hide' unwanted translations.

An Example

In the screen shot reproduced above, several translations have been specified. In particular,

```
[013][010]=[013][010]
[013]=_
[000]=
```

Now, imagine that the target file is a 'mostly' text file which consists of text lines ended by carriage return/line feed characters (CRLF, i.e. [013][010]). Unfortunately it also has several random NULL and single CR [013] characters strewn among the text. This is causing your word processor to choke when importing the text. By 'translating' the file with OttoSource, and the above substitutions, we can convert the random CR's to underscores, the NULL characters to spaces, and leave the CRLF pairs (the line endings) in place. Here's how it works.

The first translation line specifies a 'gratuitous' substitution - the CRLF pairs are translated to themselves. However, this keeps OttoSource from blindly performing the second substitution (which converts the [013] CR characters to underscores) on every CR character it encounters. When it finds a CRLF pair, it performs the no effect substitution; when it finds a single CR character, it translates it to an underscore. In this way we remove the unwanted characters without destroying the information content of the file. In other words, the first line 'blocks' the second line and helps to refine the action.

The third translation converts all NULL characters in the file to a space. (You just can't *see* the space!).

For most of us, translating control characters requires an ASCII or ANSI character chart to look up the decimal values. These are available in many DOS and Windows manuals. If you need to translate a string

of the form '[abc]', (which looks just like a control character to the OttoSource parser), you will need to specify one of the brackets using control character notation, e.g. '[091]abc'.

Printing

You can print the current file by choosing **File/Print** from the menu. You may want to preview the output using **File/Print Preview** first. The output format is based on the current line number setting as well as the configuration options for margin, header and font. OttoSource will wrap the output lines if you set the line wrap parameter in the configuration dialog. Printing operations are speeded greatly if you choose a printer device font.

If you wish to print only a range of lines, highlight a selection and choose **File/Print Selected**. Verify the result in the print preview screen, and choose **Print** to continue or **Cancel** to stop.

You can print a list of the files in the Search File window by choosing **File/Print Searchlist...** OttoSource does two things. First, a text list of the files is copied to the clipboard. Second, the list is placed in the File Viewer window and a Print Preview is initiated. You can inspect the output, continue with the print job or cancel.

Configuration

The **File/Configure** dialog lets you set various options for your viewing and printing pleasure:

- * Tab size
- * Line numbers
- * Print margin (left, in inches)
- * Wrap line column (0 disables) for printing
- * Header (pathname, date/time stamp, page number)

Tabs are expanded when the file is first read - if you need to change the setting for the current file, you will need to change the configuration, and force a reload. You can wrap printed output by specifying a wrap column > 0.

You can also choose **File/Reset Criteria** to clear all the selection criteria stored in the combo boxes.

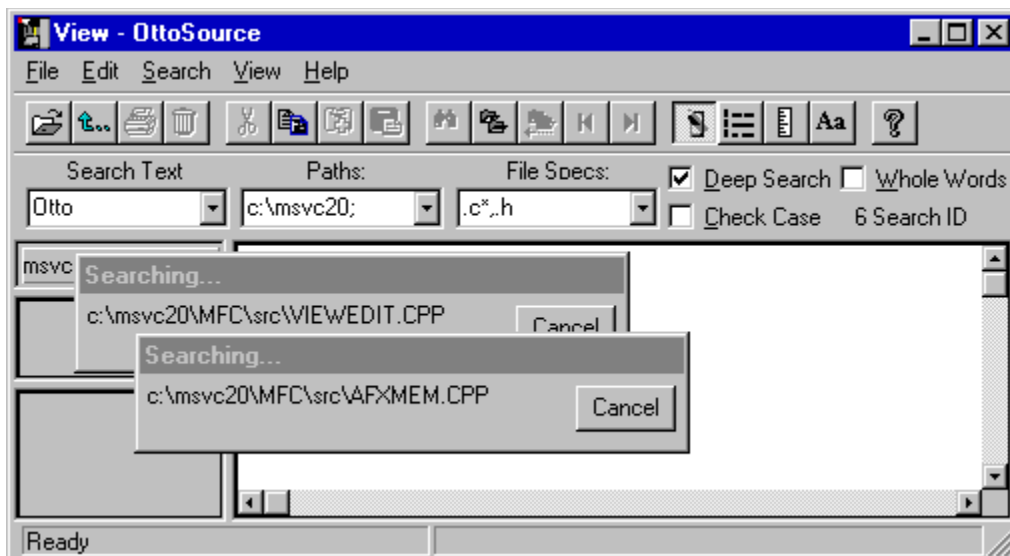
If you wish to inhibit the **File/ Delete** option, create a file in the same subdirectory as the **OTTOSRC.EXE**. The file can contain anything, but it must be named "NOKILL" (without an extension). When OttoSource starts, it checks to see if a file by that name is present , and if found it disables the option.

Multitasking Searches

OttoSource supports multithreaded searches - that is, you can launch several search requests and have them process more or less simultaneously. There is actually some gain from this. If the searches access the same directories and files, the second (and subsequent) search benefits from the file pages already in memory, dramatically reducing the overall time compared to separate, serial searching.

Of course, nothing's free, and you need to have a speedy processor and disk for this to actually be a benefit.

The snapshot below shows OttoSource executing 2 search threads.



Viewing the results

Once your search threads have completed, you can view the results in the "Search Results" window pane and choose a file to view. You can *cycle* between the different searches by clicking the **Next** and **Prior Search** buttons on the buttonbar. As you move from search to search, the original criteria are restored to the **Search Text**, **Paths** and **Ext** combo boxes and the **Search ID** text box is updated. This way you can easily tell what you were looking for.

NB: OttoSource will not close down until all the threads have completed. Choose the **Cancel** button to abort a search.

Memory Consumption

OttoSource stores the active file in memory. A large file will consume roughly as much memory as its disk size in bytes. When you close a file, or load a new one, OttoSource does NOT release the memory allocated for the prior file - it simply reuses the memory for the new one and allocates more if it needs it. If the new file is very small, this results in OttoSource holding on to the excess memory which is not in use. One advantage to this scheme is that subsequent files load more quickly since the memory has already been allocated.

In any case, OttoSource releases all memory when you quit the program.

Copyright

OttoSource is a copyrighted work of Doug Overmyer. It may be used freely in any non-commercial setting, but all other use must be approved by the author.

Disclaimer

You use OttoSource entirely at your own risk. Only you can determine its suitability for your configuration and needs. No other warranties are offered or implied.

Acknowledgements

OttoSource was inspired by an anonymous file found on CIS. I wish to thank that author. Many thanks to Günter Herbig (Hannover) and Pete Klimek for testing and suggestions.

Author:

Doug Overmyer
CIS:71021,2535
Internet:overmyer@netcom.com
MSN:doug_overmyer@msn.com

1.0 - first Win32 release. Use memory mapped files, threaded searches; many internal changes

1.1 - better multitasking behavior, deserialized thread queues.