

SCC (TM) SOURCE CODE COLORIZER

**SCC command line version
SCC Netscape Navigator Plug-In version**

December 1998

**Copyright (C) Juergen Mueller (J.M.) 1995-1998.
All Rights Reserved World-Wide.**

DISCLAIMER OF WARRANTY

THIS SOFTWARE AND ACCOMPANYING WRITTEN MATERIAL (INCLUDING INSTRUCTIONS FOR USE) IS PROVIDED "AS IS" WITH NO WARRANTIES OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THE WARRANTIES OF DESIGN, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR ARISING FROM A COURSE OF DEALING, USAGE OR TRADE PRACTICE. THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THE SOFTWARE IS WITH YOU.

THE AUTHOR AND COPYRIGHT HOLDER SHALL HAVE NO LIABILITY WITH RESPECT TO THE INFRINGEMENT OF COPYRIGHTS, TRADE SECRETS OR ANY PATENTS BY THIS SOFTWARE OR ANY PART THEREOF.

IN NO EVENT WILL THE AUTHOR AND COPYRIGHT HOLDER BE LIABLE FOR DAMAGES, INCLUDING ANY LOST PROFITS, LOST MONIES, OR OTHER DIRECT, INDIRECT, GENERAL, SPECIAL, INCIDENTAL, EXEMPLARY OR CONSEQUENTIAL DAMAGES ARISING IN ANY WAY OUT OF THE USE OR INABILITY TO USE THIS PROGRAM (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, BUSINESS INTERRUPTION, LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), MODIFICATION, DISTRIBUTION AND ON ANY THEORY OF LIABILITY, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY.

NO DEALER, AGENT, OR EMPLOYEE OF LICENSOR IS AUTHORIZED TO MAKE ANY MODIFICATIONS, EXTENSIONS, OR ADDITIONS TO THIS LIMITED WARRANTY.

ACKNOWLEDGMENT

BY USING THIS SOFTWARE YOU ACKNOWLEDGE THAT YOU HAVE READ THIS LIMITED WARRANTY AND ACCOMPANYING REMARKS, UNDERSTAND IT, AND AGREE TO BE BOUND BY ITS TERMS AND CONDITIONS. YOU ALSO AGREE THAT THIS IS THE COMPLETE AND EXCLUSIVE STATEMENT OF AGREEMENT BETWEEN THE PARTIES AND SUPERSEDE ALL PROPOSALS OR PRIOR AGREEMENTS, ORAL OR WRITTEN, AND ANY OTHER COMMUNICATIONS BETWEEN THE PARTIES RELATING TO THE SUBJECT MATTER OF THE LIMITED WARRANTY. YOU AGREE THAT THIS NOTICE APPLIES TO ALL FILES IN THIS SOFTWARE DISTRIBUTION.

You are expressly prohibited from selling this software or parts of it in any form, circulate it in any incomplete or modified form, distribute it with another product (except as Shareware) or removing this notice. No one may modify or patch any of

the executable files in any way, including, but not limited to, decompiling, disassembling or otherwise reverse engineering this software in whole or part.

The documentation may be distributed verbatim, but changing is not allowed. The information and specifications in this document are subject to change without notice.

THIS VERSION OF THE DOCUMENTATION, SOFTWARE AND COPYRIGHT SUPERSEDES ALL PREVIOUS VERSIONS. THIS SOFTWARE AND ITS RELATED DOCUMENTATION MAY CHANGE WITHOUT NOTICE. THIS DOCUMENTATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN, THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THE DOCUMENTATION. THE AUTHOR MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENTATION AT ANY TIME.

This software and documentation is Copyright (C) 1995-1998 by

**Juergen Mueller
Eastleighstrasse 14
70806 Kornwestheim
GERMANY**

Email: jmsxt@compuserve.com

Homepage: <http://ourworld.compuserve.com/homepages/jmsxt>

THERE ARE NO AND HAVE NEVER BEEN RELATIONS BETWEEN THE AUTHORS PROFESSIONAL WORK AND THE SXT DEVELOPMENT. THE SXT PROJECT IS AND HAS EVER BEEN AN INDEPENDENT PRIVATE PROJECT OF THE AUTHOR AND IS WRITTEN AND MAINTAINED IN HIS FREE TIME.

LICENSE

This software is not public domain or free software, but is being distributed as shareware. It is protected by copyright and distributed under this license restricting its use.

Non-registered users of this software are granted a limited license for a 30-day evaluation period starting from the day of the first use to make an evaluation copy for trial use for the express purpose of determining whether this software is suitable for their needs. At the end of this trial period you should either register your copy or discontinue using this software. The use of unregistered copies of this software, outside of the initial 30-day trial, by any person, business, corporation, government agency or any other entity is strictly prohibited.

This means that if you use this software, then you should pay for your copy. This software is not free, but you have the opportunity to try it before you buy it. Either pay for it, or quit using it. A registration entitles you to use your copy of this software in its actual version on any and all computers available to you. If other people have access to this software or may use it, then additional copies or a site license should be purchased. If you want to use a newer version of this software you have to license it.

All users are granted a limited license to copy this software only for the trial use of others and subject to the above limitations. This license does not include distribution, selling or copying of this software package in connection with any other product or service or for distribution in any incomplete or modified form. Operators of electronic bulletin board systems and software servers (like Internet World Wide Web (WWW) Sites or FTP-Servers) are encouraged to post this software for downloading by their users, as long as the above conditions are met.

This package is expected to be distributed as shareware for download, on disk or on CD-ROM, but the fees paid for "distribution" costs (disk, CD-ROM) are strictly exchanged between the distributor and the recipient, and the author makes no express or implied warranties about the quality or integrity of such indirectly acquired copies. Distributors and users may obtain the package directly from the author by following the ordering procedures in the REGISTER files.

REGISTRATION REMINDER

Unregistered copies of this software are 100% fully functional. I make them this way so that you can have a real look at them, and then decide whether they fit your needs or not. This work depends on your honesty. If you use it, I expect you to pay for it. When you pay for the shareware you like, you are voting with your pocketbook, and will encourage me and other shareware authors to develop more of these kinds of products.

THANK YOU FOR SUPPORTING THE SHAREWARE CONCEPT

TABLE OF CONTENTS

1 SCC SOURCE CODE COLORIZER	6
2 PROGRAM DESCRIPTION	6
2.1 OVERVIEW	6
2.2 SUPPORTED LANGUAGES	8
2.3 KNOWN PROBLEMS	9
3 SCC OPTIONS	11
4 SCC PLUG-IN OPTIONS	20
5 REFERENCES	22
6 APPENDIX	24
6.1 APPENDIX 1: SYSTEM REQUIREMENTS	24
6.2 APPENDIX 2: INSTALLATION	24
6.3 APPENDIX 3: AVAILABILITY	25
6.4 APPENDIX 4: TRADEMARKS	25

1 SCC SOURCE CODE COLORIZER

The SCC Source Code Colorizer is a powerful program utility that supports source code analysis, maintenance and documentation. SCC helps programmers in publishing source code on the Internet in an impressive way. SCC reads source files and generates HTML output with a colored copy of that source code. The resulting HTML output can be viewed with WWW-Browsers like Netscape Navigator / Communicator, Microsoft Internet Explorer and others. SCC supports more than 20 popular programming languages like C, C++, Objective-C, Java, Fortran, dBase, Clipper, Pascal, Ada, Lisp, Cobol, Modula, Basic, PL/1 and others. Besides the stand-alone SCC command line version there is also a Netscape Navigator Plug-In version available which integrates the SCC functionality directly into Netscape Navigator under Windows 95 / Windows NT and LINUX.

SCC is another part of the SXT Software Exploration Tools, a collection of programming tools which provide various source code analysis capabilities for many different programming languages. The following SXT packages are available:

- SCC - Source Code Colorizer**
- CXT - C Exploration Tools**
- DXT - DBASE Exploration Tools**
- FXT - FORTRAN Exploration Tools**
- JXT - JAVA Exploration Tools**
- LXT - LISP Exploration Tools**

2 PROGRAM DESCRIPTION

2.1 OVERVIEW

The SCC Source Code Colorizer generates HTML output which is a colored version of the used source file. In the HTML output all keywords, comments, strings, character constants, numbers, library functions, parentheses ('(')'), braces ('{' '}'), brackets ('[' ']'), separators ('.', ':', ';'), delimiters and preprocessing tokens are marked with specific colors and font styles to make source code reading easier. The font colors and font styles as well as the background color can be changed by the user. There are additional options to customize the resulting output like inserting line numbers.

The implementation of the SCC source code analysis capabilities for each of the supported programming languages was mainly based on available standards, reference books and online help information. For some programming languages this information is very clear and can be used without any problems. Other programming languages especially those which exist in several variants like Lisp, Scheme, Pascal, Assembler, Basic, dBase or Clipper are more problematic. For the SCC implementation it was tried to cover as many features of these programming language as possible.

SCC exists in different versions, as a command line version for 16 bit DOS (SCC16.EXE), 32 bit DOS Protected Mode (SCC386.EXE), 32 bit Windows NT / Windows 95 (SCC.EXE), OS/2 (SCC4OS2.EXE) and LINUX (SCC) and as a 32 bit Netscape Navigator / Communicator Plug-In for Windows NT / Windows 95 (NPSCC32.DLL) and LINUX (NPSCC.SO).

The command line versions can be customized with various command line options which are described later in this document. The 32 bit SCC Windows NT / Windows 95 / OS/2 / LINUX versions write the resulting HTML output for source file 'xxx.yyy' into file 'xxx.yyy.HTML' (long filenames, extension '.HTML' is appended). The 16 bit SCC DOS version replaces the existing file extension with '.HTM' (DOS 8.3 filenames), the resulting HTML output from 'xxx.yyy' is named 'xxx.HTM'.

There are no checks whether HTML-files with equal names do already exist, therefore such files will always be overwritten without warning. If problems are encountered during processing, SCC gives short warning messages and tries to synchronize itself to continue the processing. However, SCC is not a programming language syntax checker and cannot find errors in the source files. Incorrect programs may cause SCC to crash.

The SCC Netscape Navigator Plug-In cannot be directly customized via command line parameters. Instead it uses either special options specified within the <EMBED> statement or specified with a command file named 'NPSCC.CMD' which must be placed in the Windows directory (usually C:\WINDOWS). NPSCC.CMD can contain all described SCC options to customize the SCC behavior, this command file processing is similar with declaring option '\$NPSCC.CMD' for the SCC command line versions. The SCC Plug-In writes the processed HTML output directly into a new browser window, there is no intermediate HTML-file and the contents of the browser window cannot be saved. Customization via the NPSCC.CMD file is currently not supported in the LINUX version

Associated program source files can also be declared with the <EMBED> HTML statement. With this it is possible to declare special options for the SCC Plug-In. For more information see the specific chapter later in this document.

The SCC Plug-In does not display detailed warnings if problems occur, only a message box opens to inform the user after SCC processing is finished. During processing SCC displays progress information in the Netscape Navigator status bar.

For best viewing results of the HTML files and the Plug-In output choose a fixed size font for the browser window.

2.2 SUPPORTED LANGUAGES

SCC supports more than 20 programming languages. The following list gives an overview about these programming languages and shows the SCC command line parameter to identify them.

Language	SCC Language Parameter
– C	C
– C++	C++
– Objective C	OBJECTIVEC
– Java	JAVA
– JavaScript	JAVASCRIPT
– Borland Pascal	BORLANDPASCAL
– ISO 7185 Pascal	PASCALISO7185
– ISO 10206 Pascal	PASCALISO10206
– Delphi	DELPHI
– Fortran 77	FORTTRAN77
– Fortran 90	FORTTRAN90
– dBase (III, III plus, IV, V)	DBASE
– Clipper (87, 5.0)	CLIPPER
– Ada 83	ADA83
– Ada 95	ADA95
– Macro Assembler	MASM
– Turbo Assembler	TASM
– GNU 386 Assembler	GAS386
– Common Lisp	LISP
– MIT Scheme	SCHEME
– Modula 2	MODULA2
– Modula 3	MODULA3
– Oberon	OBERON
– Cobol 74	COBOL74
– Cobol 85	COBOL85
– PL/1	PL1
– SQL	SQL
– Quick Basic	QBASIC
– Visual Basic	VBASIC
– Eiffel	EIFFEL

The language parameter can be used to override the default language which is determined by the source filename extension. It must also be used with the '.ext=lang' option.

By default, the programming languages of the source files are recognized by the file extensions. Due to possible conflicts with file extensions which are used for several languages or where different variants of one programming language exist, the following listed default values are used.

Language	Default Filename Extension
– C	.C
– C++	.CPP, .CC, .CXX, .C++, .H, .HPP, .HH, .HXX, .INC (*)
– Objective C	.M
– Java	.JAVA, .JAV, .J
– JavaScript	.JS, .JSCRIPT, .LS, .LSCRIPT
– Borland Pascal	.PAS, .P, .PASCAL
– Delphi	.DPR
– Fortran 77	.F, .FOR, .F77
– Fortran 90	.F90
– Clipper	.PRG
– Ada 95	.ADB, .ADS, .ADA, .A
– Macro Assembler	.ASM
– GNU Assembler	.S
– Common Lisp	.LISP, .LSP, .EL, .ELISP
– MIT Scheme	.SCM, .SS
– Modula 2	.MOD (**), .MD, .MI, .M2
– Modula 3	.M3
– Oberon	.MOD (**), .DEF
– Cobol 85	.CBL, .COB, .COBOL
– PL/1	.PL1
– SQL	.SQL
– Visual Basic	.BAS, .BASIC
– Eiffel	.E, .EIF

(*) Source files with the extensions .H, .HPP, .HH, .HXX, .INC are all handled as C++ source code. Even for C header files (extension .H) this assumption should produce acceptable results.

(**) To distinguish Modula 2 and Oberon source files both with extension .MOD, SCC looks if the source file has a binary header (first WORD is 0xF001 or 0x01F0) which is typical for Oberon. If no such header is found the source file is handled as a Modula 2 source file. If Oberon source files without a binary header are present it is necessary to use option `-.ext=lang`.

Languages which are not listed here are not automatically recognized and must be specified either via option `-language` with 'language' as the desired SCC language parameter or via option `-.ext=lang`. See the SCC options description for more information.

2.3 KNOWN PROBLEMS

HTML CHARACTER SET

If source code contains special characters (semi-graphic, non ISO characters), these characters are also in the resulting HTML output. It depends on the browser how

such characters are displayed. HTML is usually based on the ISO 8859 Latin character set.

FILE RECOGNITION WITH NETSCAPE NAVIGATOR PLUG-IN

Under some currently unknown circumstances source files are not transferred to the SCC Netscape Navigator Plug-In, although they are correctly recognized by Netscape Navigator as an SCC Plug-In associated file. The result is either a directly loaded source file or a new empty browser window. However, it can also happen that the SCC Netscape Navigator Plug-In will be invoked with a file not associated with it e.g. an HTML-file. Both cases can happen with files on the local computer hard disk and with files that are loaded via a network connection from other machines (WWW, FTP). The reasons for this strange behavior is not known and it can usually not be repeated.

NETSCAPE NAVIGATOR PLUG-IN CRASH

If the Netscape Navigator window for the SCC Plug-In output is closed before the processing and writing is completely finished, Netscape Navigator will crash because the SCC Plug-In does not recognize this situation. This is especially the case for very large source files. To inform the user a progress information is written to the status bar of the Netscape Navigator window where the SCC Plug-In was started from.

RECOGNITION OF FLOATING POINT NUMBERS

Floating point numbers are not correctly recognized. Especially leading dots '.', exponential indicators 'E' and negative signs '-' may not be handled as they should and will probably have wrong colors.

3 SCC OPTIONS

This section gives a complete overview about all SCC options and their syntax. It gives also remarks for their use and shows several examples with detailed descriptions. The options are case-sensitive! This section of the documentation should be read very careful by all users to get an overview about all the features which are provided.

THE OPTIONS ARE LISTED IN LEXICOGRAPHICAL ORDER. NONE OF THESE OPTIONS IS SET BY DEFAULT.

SYNTAX: **SCC** [options [\$cmdfile]] <[+]file> <@filelist>

OPTIONS:

-ADA83

Assume source files with programming language Ada 83.

-ADA95

Assume source files with programming language Ada 95.

-BORLANDPASCAL

Assume source files with programming language Borland Pascal.

-C

Assume source files with programming language C.

-C++

Assume source files with programming language C++.

-COBOL74

Assume source files with programming language Cobol 74.

-COBOL85

Assume source files with programming language Cobol 85.

-CLIPPER

Assume source files with programming language Clipper (Clipper 87, Clipper 5.0).

-DBASE

Assume source files with programming language dBase (dBase III, dBase III plus, dBase IV, dBase V).

-DELPHI

Assume source files with programming language Borland Delphi.

-DIRpath

Set the output directory to 'path'. By default, the resulting HTML files are written to the same directory as the source files.

-Echarset

Extend the DEFAULT character for identifier recognition with a user defined character set 'charset'. All characters must be specified within one -E option. This option allows the programmer to use national character sets as they are common in Germany, Denmark, Sweden and other European countries.

-EIFFEL

Assume source files with programming language Eiffel.

-FORTRAN77

Assume source files with programming language Fortran 77.

-FORTRAN90

Assume source files with programming language Fortran 90 (free format).

-GAS386

Assume source files with programming language GNU 386 Assembler.

-H[elp]

See option -?.

-JAVA

Assume source files with programming language Java.

-JAVASCRIPT

Assume source files with programming language JavaScript.

-KEYW@file

Specifies a 'file' with additional keywords. The built-in programming language specific set of keywords can be extended with this option.

-L[L][+]

Redirect the screen output to a file, called 'SCC.LOG'. If '+' is set, the output is both written to screen and redirected to the log file so that the output messages can both be viewed as they appear and later analyzed. Finally, -LL resp. -LL+ appends the output to an existing log-file, this can be useful if SCC is run from a batch-file to catch all output.

-LIB@file

Specifies a 'file' with names of library functions. The displaying of library functions can be customized in the font description file with the 'color_library' and 'style_library' parameters (see option -font@file). By default, there are no library functions defined in SCC. To use the library function coloring features the user has to specify a list with names of library functions.

-LISP

Assume source files with programming language Common Lisp. SCC should also be able to handle other LISP dialects.

-MASM

Assume source files with programming language Microsoft Macro Assembler.

-MODULA2

Assume source files with programming language Modula 2.

-MODULA3

Assume source files with programming language Modula 3.

-N

Disable output file writing. This option can be useful if the source files shall only be checked for errors.

-NOCPPCMT

Do not accept C++ comments '/*...*/' in C source code. By default, C++ comments are recognized in C code since this is now a common feature of many popular compilers like Microsoft, Borland or GNU and will probably be also in the next C standard.

However, this option can be used to ensure compatibility with C compilers which do not recognize C++ comments within C source code.

-OBERON

Assume source files with programming language Oberon. The Oberon specific binary source file headers are recognized and handled.

-OBJECTIVEC

Assume source files with programming language Objective C.

-PASCALISO7185

Assume source files with programming language ISO 7185 Pascal (Standard Pascal).

-PASCALISO10206

Assume source files with programming language ISO 10206 Pascal (Extended Pascal).

-PL1

Assume source files with programming language PL/1.

-QBASIC

Assume source files with programming language Quick Basic.

-SCHEME

Assume source files with programming language MIT Scheme. SCC should also be able to handle other SCHEME dialects.

-SQL

Assume source files with programming language SQL.

-TASM

Assume source files with programming language Borland Turbo Assembler.

-VBASIC

Assume source files with programming language Visual Basic.

-font@file

Specifies a 'file' with font color and style description. This option overrides the default color and style values. An SCC font file ('SCCFONT') with the SCC default values is provided as an example.

The following color definition values are available:

- color_background
- color_comment
- color_keyword
- color_library
- color_preprocessor
- color_string
- color_character
- color_number
- color_parentheses
- color_braces
- color_brackets
- color_separator
- color_delimiter

Colors are defined with `color_... = #rrggbb`, where `#rrggbb` is the HTML syntax for color description (rr = red, gg = green, bb = blue). The colors range from #000000 (= black) to #FFFFFF (= white).

The following font style definition values are available:

- style_comment
- style_keyword
- style_library
- style_preprocessor
- style_string
- style_character
- style_number
- style_parentheses
- style_braces
- style_brackets
- style_separator
- style_delimiter

Font styles are defined with `style_... = style`, where `style` is either I (= italics), B (= bold), U (= underline), S (= strike through), a combination of them or none. This syntax is based on the HTML font style syntax. For a more detailed description of the HTML font description capabilities see the HTML reference.

This option gives the users the greatest flexibility to customize the resulting HTML output for their needs.

-h[elp]

See option -?.

-l

Recognize language specific extensions that are not handled by default. This affects the programming languages Cobol and Fortran where additional keywords are enabled.

-lineno

Insert line numbers into the HTML output. Very useful for navigating and exactly identifying source code locations.

-.ext=lang

Associate source language 'lang' with file extension '.ext'. This option overrides the initial association with a default language for file extensions. The modifier 'lang' must be one of the given language options (C, C++, OBJECTIVEC, JAVA, ...). Example: '.MOD=OBERON' associates extension '.MOD' with programming language 'OBERON' and overwrites therefore the default association with 'MODULA2'.

-\$

Recognize \$ as identifier character, not as delimiter. Useful for FORTRAN source code.

-?

Shows the command line syntax and gives a short, but complete help information about the accepted commands and their syntax.

COMMAND LINE FILES**cmdfile**

Specifies a file with (additional) command line options. This might be useful if the command line would be too long because of the number of options and files declared or if you are usually using the same options which can then be stored in a command file. The initial '\$'-character is required to mark a command file. Inside a command file it is possible to put comment lines, they start with a '#' sign in the first column and are skipped til end-of-line.

filelist

A file with a list of source file(s) to be processed, wildcards are accepted. The list file should have every file on a single line. The rules for files containing assembler code and path translation are described above. The initial '@'-character is required to

mark a file list file. The '+' sign for subdirectory processing is also possible inside the file list file. Inside a file list file it is possible to put comment lines, they start with a '#' sign in the first column and are skipped til end-of-line.

[+]file

The name of a source file to be processed. More than one file can be specified on the command line. The '+' sign indicates that, starting from the given directory, all sub-directories should be searched recursively for the given filename search pattern. This addition is useful if a large software project is divided into several modules with separate sub-directories for each module. In that case only the starting (root-) directory with the requested filename search pattern must be specified to search the current directory and all sub-directories.

If the filename or the include file specification inside a file contains a relative path ('./', '..\', '../' or '..\') it will be translated into an absolute path starting from the current working directory respectively in case of include files depending on the path of the parent file. Command line wildcards '*' and '?' are possible and will be accepted.

REMARKS ON USING OPTIONS

NONE OF THE DESCRIBED OPTIONS IS PREDEFINED SO IT IS UP TO THE USER HIMSELF TO CUSTOMIZE HIS PREFERRED PROCESSING BEHAVIOR AND OUTPUT STYLE BY ADDING CONTROL OPTIONS NEEDED THEREFORE.

This assumption seems to be the best way to give users the freedom of making their own decisions about the features they really need for doing their work, although the large number of options may be confusing for beginners. Therefore, take some time to learn about SCC and its features, read this manual carefully and make your own experience with this software.

It is possible to declare more than one source file, command file and list file on the command line. In that case they will be processed in the order they appear. Files and options can be placed in mixed order on the command line, there is no recommended order for them because all options (also those inside command files!) will be processed before any source files are scanned.

The maximum command line length for DOS is 127 characters, so this is a system dependent 'natural' limit for the options and filenames being declared. If you have more items to declare, place them into command list files and file list files, which do not have such limitations.

Options can also be defined by the environment variable SCC (not applicable for SCC Plug-In version) like

SET SCC=...

To separate single options in the environment string, spaces are required.

The rules for the interpretation of options is

1. if defined, all options in the environment variable SCC will be taken,
2. the command line options and the option files will be interpreted in the order they appear.

If an option is declared different more than once then previous declarations will be overwritten by the newer one.

Filenames being composed of drive letter, directory name, filename and file extension, in the following referred simply as 'pathname', are treated by some special procedures to force a unique style of their internal representation:

- path names are considered case insensitive under DOS, Windows 95 / Windows NT and OS/2, so there is no difference in upper case / lower case / mixed case path names, the path names remain unchanged as they are specified by the user or defined within source files,
- path names are considered case sensitive under LINUX, so that it makes a difference to use upper or lower case characters for path names (a file search for 'c*.c' is different to 'C*.c'), the path names remain unchanged as they are specified by the user or defined within source files,
- path names containing './', '\.', '..\' and '..\' (so called 'relative paths') are expanded and transformed into absolute paths,
- the recommended directory delimiter is '/' (UNIX-style), if a '\' (DOS-style) is recognized in a path name, it will be replaced by '/',
- path names are always expanded and transformed into the default style

<DRIVE LETTER>:<DIRECTORY PATH>/<FILENAME>

to get a unique representation for every filename that must be handled during processing,

- filenames are operating system dependent:
 - DOS: maximum length: 12 characters (8.3 format)
 - Windows 3.1: maximum length: 12 characters (8.3 format)
 - Windows NT: maximum length: 255 characters
 - Windows 95: maximum length: 255 characters
 - OS/2: maximum length: 255 characters
 - LINUX: maximum length: 255 characters(for Windows 95 / NT, OS/2 and LINUX, the complete path name itself, including the filename, can have a maximum of 255 characters)

COMMAND LINE EXAMPLES

1. SCC hello.c object.cpp driver.asm

SCC processes in the current directory a C file (hello.c), a C++ file (object.cpp) and a MS Macro Assembler file (driver.asm).

2. SCC +*.c +*.cpp

SCC32 processes all files with extension '.c' and '.cpp' in the current directory and all sub-directories (recursive search). The default languages are used, C for extension '.c' and C++ for extension '.cpp'.

3. SCC -.mod=OBERON *.mod

SCC processes all files with extension '.mod' in the current directory. Extension '.mod' is associated with the programming language OBERON. This overrides the default SCC association which is MODULA2.

4. SCC -.prg=DBASE -font@MYFONT -DIRc:\temp *.prg

SCC processes all files with extension '.prg' in the current directory. Extension '.prg' is associated with the programming language DBASE. This overrides the default SCC association which is CLIPPER. The color and font description is given in the file MYFONT. The resulting HTML output files are written to directory c:\temp.

5. SCC -ADA95 -lineno *.a*

SCC processes all files with extension '.a*' in the current directory and assumes ADA95 source code. Line numbers are inserted into the generated HTML-output files.

6. Example for NPSCC.CMD

Following is an example for the contents of a typical SCC Netscape Navigator Plug-In command file NPSCC.CMD (which must be placed in the Windows directory):

```
# command file
-lineno
-.ada=ADA83
-font@c:\scc\sccfont
```

After a comment line starting with '#' options for line numbering, file extension association and font description are given. These options are used for every file that is processed with the SCC Plug-In. This command file can also be used from the SCC command line versions (assuming c:\windows as the Windows directory) with

```
SCC $c:\windows\npscc.cmd
```

and via environment variable SCC with

```
set SCC=$c:\windows\npscc.cmd
```

and allows a consistent use of options with all SCC versions.

4 SCC PLUG-IN OPTIONS

With the <EMBED> HTML statement it is possible to declare SCC associated program source files within an HTML document and to specify some additional special options for the SCC Plug-In. The EMBED statement with all possible options for the SCC Plug-In looks as follows:

**<EMBED SRC="filename" LINE=linenumber LINENO=1
TARGET=targetwindow HIDDEN=true>**

where SRC and HIDDEN are HTML specific parameters for <EMBED>. SRC declares the filename for which an associated Plug-In shall be invoked and HIDDEN defines whether there is a Plug-In area on the page or not. Only option SRC to define the file name is mandatory, all other <EMBED> options are optional.

The SCC specific <EMBED> options are

-LINE=linenumber

With option LINE=linenumber (linenumber >= 0) it is possible to declare a target line number in the source code which shall be initially displayed in the Netscape Navigator window. By default line number 0 is assumed that positions the HTML output to the beginning of the source file.

-LINENO=1

With option LINENO=1 it is possible to enable line numbering in the HTML output. By default no line numbers are inserted.

-TARGET=targetwindow

With option TARGET=targetwindow it is possible to specify the name of a target window for the HTML output. Targetwindow can be an existing window, a frame name specified in a <FRAMESET> tag or it can be one of the literal frame names _top, _parent, _self, or _blank. By default the HTML output is loaded into a new and empty ("_blank") Netscape Navigator window.

Besides these options the previously mentioned NPSCC.CMD command file is additionally used.

With the following JavaScript function it is possible to handle different parameters for the SCC Plug-In and to provide an alternative behavior for situations where the SCC Plug-In is not installed.

```
<SCRIPT>
<!--
function vsrc(fname,fline)
{
    var SccPlugIn = navigator.plugins["SCC"];
    if (SccPlugIn)
        document.write('<EMBED SRC="' + fname + '" line=' +
```

```
                                fline + ' lineno=1 HIDDEN=true>')
    else
        window.open(fname)
    }
    // -->
</SCRIPT>
```

In the HTML source, the script function can be invoked with a HREF link declared as

```
<A HREF=" " onClick="vsrc('file:///path/filename',line)">...</A>
```

By clicking on that link the function `vsrc()` will be called and the Plug-In, if installed, will be invoked.

5 REFERENCES

Brian W. Kernighan, Dennis M. Ritchie: "The C Programming Language", Prentice Hall, Englewood Cliffs, Second Edition 1988

Samuel P. Harbison, Guy L. Steele Jr.: "C: A Reference Manual", Prentice Hall, Englewood Cliffs, Third Edition 1991

Bjarne Stroustrup: "The C++ Programming Language", Addison-Wesley, Second Edition 1992

"Working Paper for Draft Proposed International Standard for Information Systems - Programming Language C++", AT&T, ANSI committee X3J16, ISO working group WG21, April 28, 1995

H. Wehnes: "FORTRAN 77", 3. Auflage, Carl Hanser Verlag, 1984

H. Sigl: "dBase/Foxbase/Clipper - Globalreferenz", Addison Wesley, 1989

Larisch, D.: "Das grosse Buch zu Clipper 5.0", Data Becker, 1991

Borchers, W.; Dilger, M.; Vonhoegen, H.: "Das grosse Buch zu dBase 5 für Windows", Data Becker, 1994

Hanna, G.: Ada 83 LRM HTML-Hypertext Reference Manual (available online at <http://www.saic.com/~gregor>)

Kempe, M.: Ada 95 HTML-Hypertext Reference Manual: Hyper-RM95 v1.0 (available online at <http://lglwww.epfl.ch/Ada/> or <ftp://lglftp.epfl.ch/pub/Ada/HTML>)

Bertrand Meyer: "Eiffel: The Reference, ISE Technical Report TR-EI-41/ER", version 3 updated by Neil Wilson (available online via ftp)

Gosling, J.; Joy, B.; Steele, G.: "The Java Language Specification"

"MIT Scheme Reference" Edition 1.32 alpha (available online via ftp)

"Common Lisp HyperSpec", The Harlequin Group Limited (available online via <http://www.harlequin.com/>)

"Pascal ISO/IEC 7185"

"Extended Pascal ISO/IEC 10206"

Larkin, D.; Wilson, G.: "Object-Oriented Programming and the Objective-C Language", NeXT Software Inc. (available online via ftp)

H. Moessenboeck, N. Wirth: "The Programming Language Oberon" (available online via ftp)

R. J. Sutcliffe: "Modula-2: Abstractions for Data and Programming Structures (Using ISO-Standard Modula-2)" (available online via ftp from ftp.twu.ca)

"Modula-3: Language Definition" (available online via ftp from src.dec.com)

W.-M. Kaehler: "Einfuehrung in die Programmiersprache COBOL", Vieweg Verlag, 4. Auflage 1988.

A. Berger, A. Beringer: "Programmieren in PL/1", Oldenbourg Verlag.

H. Schliessmann: "PL/1 fuer Mikrocomputer", BI Wissenschaftsverlag.

M. Bohl, A. Walter: "PL/1", Oldenbourg Verlag.

K.-H. Hauer: "PL/1 - Ein Lehr- und Arbeitsbuch fuer die Praxis".

R. Habib: "COBOL Workbench professional", IWT Verlag 1996.

ISO/IEC 9075; Database Language SQL, 1992.

6 APPENDIX

6.1 APPENDIX 1: SYSTEM REQUIREMENTS

16 bit version:

- IBM-AT or 100% compatible with Intel 80286 or higher, 640 KB RAM, hard-disk, MS-DOS 5.0 or higher

32 bit version:

- IBM-AT or 100% compatible with Intel 80486 or higher, 16 MB RAM, hard-disk, Windows NT 3.51, Windows NT 4.0 or Windows 95, OS/2 3.0 resp. LINUX Kernel 2.0.32

6.2 APPENDIX 2: INSTALLATION

SCC Command line version (SCC16.EXE, SCC.EXE, SCC4OS2.EXE, SCC):

Copy the EXE-file into your utility directory (like C:\UTIL) and ensure that this directory is included into the PATH environment variable.

SCC Netscape Navigator Plug-In:

Windows 95 / Windows NT: Copy the SCC Netscape Navigator Plug-In DLL-file (NPSCC32.DLL) into the Netscape Navigator Plug-In directory '.\plugins'. This directory is a subdirectory of the Netscape Navigator program directory (for Windows usually C:\Netscape\Navigator\Program\plugins). On Windows, installed plug-ins are automatically configured to handle the MIME types that they support.

LINUX: Copy the SCC Netscape Navigator Plug-In shared object file (NPSCC.SO) into the Netscape Navigator Plug-In directory './plugins'. This directory is a subdirectory of the Netscape Navigator program directory (for LINUX usually /usr/local/netscape/plugins). Then you must manually install the plug-in and configure the supported MIME types. On LINUX, the user can flexibly configure which MIME types are handled by which plug-ins or helper applications, by using the "Navigator" -> "Applications" dialog of the "Edit" -> "Preferences" window. "Applications" displays a list of all handled MIME types and whether the type is handled by Navigator, a plug-in, or a helper application. The user can select any MIME type in the list and click the "Edit..." button to change how the MIME type is handled, including selecting a different plug-in to handle the type from a pop-up menu of all plug-ins that support that type. For SCC, simply choose the SCC plug-in from the pop-up menu (if already present) to get the supported suffixes or choose the "New" button and type the description "SCC Source Code Colorizer" and MIMEType "application/x-scc" manually into the dialog field. Then in the scroll box near "Plug In:" the string "SCC" should appear. Select this choice and the supported suffixes should be installed into the dialog. Finally, be sure to set the read / write / execution rights (rwx) correctly, because the NPSCC.SO plug-in must be able to

create, open and delete files during its normal operation, even for a normal user and not only for super user.

6.3 APPENDIX 3: AVAILABILITY

The SHAREWARE version of the SCC can be downloaded via WWW and FTP from many Internet sites.

The following list is not complete, only the known WWW- and FTP-sites (usually primary sites where the software is uploaded by the author or which are known to be up-to-date) are shown.

WWW / FTP:

The primary upload site for all SXT programs is SIMTEL.NET:

WWW: <http://www.simtel.net/simtel.net>

FTP: [ftp.simtel.net](ftp://ftp.simtel.net)

6.4 APPENDIX 4: TRADEMARKS

All brand or product names are trademarks (TM) or registered trademarks (R) of their respective owners.

The following products and names are Copyright (C) Juergen Mueller (J.M.), all rights reserved world-wide:

SCC (TM) SOURCE CODE COLORIZER

This package is part of

SXT (TM) SOFTWARE EXPLORATION TOOLS

a collection of several programs for the source code analysis of many different programming languages.