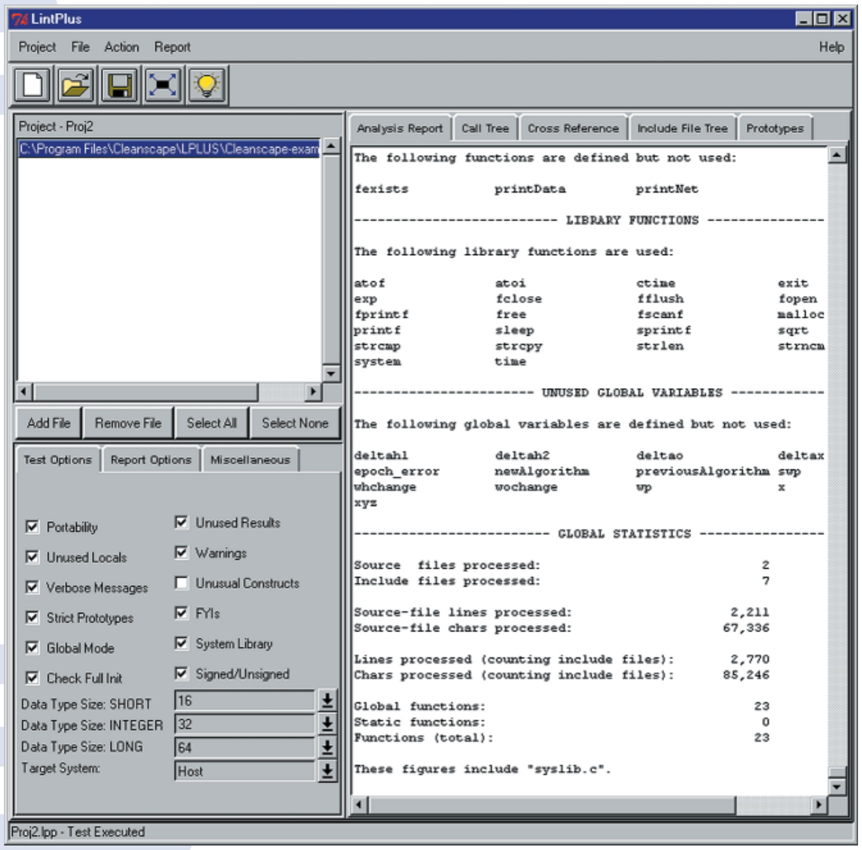
CLEANSCAPE

Cleanscape LintPlus 5

Static Source Code Analyzer for C

Key Features & Benefits

- Catches problems at the source code level
- Speeds debugging
- Facilitates maintenance of large C applications
- Enforces coding standards within a programming group
- Increases application portability
- Maps out unfamiliar programs
- Detects bugs that typically pass through compilers
- Documents code
- Automatically generates call trees & cross-references
- Intuitive graphical interface **New!**
- Advanced GNU support **New!**
- Enhanced embedded environments support — including a Windows host. **New!**
- Expanded reporting options **New!**
- Cross-platform support: Unix, Linux, Windows **New!**



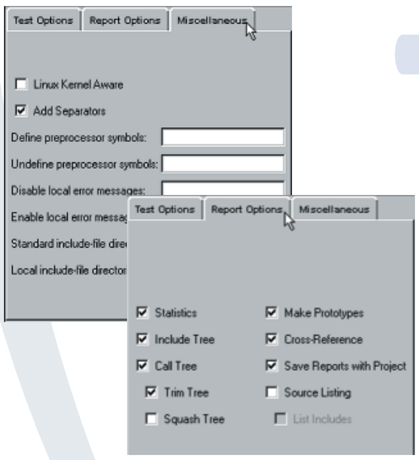
Test Options:

- [✓] Portability
- [✓] Unused Locals
- [✓] Verbose Messages
- [✓] Strict Prototypes
- [✓] Global Mode
- [✓] Check Full Init
- [✓] Unused Results
- [✓] Warnings
- [ ] Unusual Constructs
- [✓] FYIs
- [✓] System Library
- [✓] Signed/Unsigned

Data Type Size: SHORT	16
Data Type Size: INTEGER	32
Data Type Size: LONG	64
Target System:	Host

```
The following functions are defined but not used:

fexists             printData           printNet

------------------------- LIBRARY FUNCTIONS ---------------

The following library functions are used:

atof                atoi                ctime               exit
exp                 fclose              fflush              fopen
fprintf             free                fscanf              malloc
printf              sleep               sprintf             sqrt
strcmp              strcpy              strlen              strncm
system              time

---------------------- UNUSED GLOBAL VARIABLES ------------

The following global variables are defined but not used:

deltahl             deltah2             deltao              deltax
epoch_error         newAlgorithm        previousAlgorithm   swp
whchange            wochange            wp                  x
xyz

------------------------ GLOBAL STATISTICS ----------------

Source  files processed:                           2
Include files processed:                           7

Source-file lines processed:                   2,211
Source-file chars processed:                  67,336

Lines processed (counting include files):      2,770
Chars processed (counting include files):     85,246

Global functions:                                 23
Static functions:                                  0
Functions (total):                                23

These figures include "syslib.c".
```

Proj2.lpp - Test Executed



Miscellaneous:

- [ ] Linux Kernel Aware
- [✓] Add Separators

Define preprocessor symbols:
Undefine preprocessor symbols:
Disable local error messages:
Enable local error messag
Standard include-file dire
Local include-file directori

Report Options:

- [✓] Statistics
- [✓] Include Tree
- [✓] Call Tree
- [✓] Trim Tree
- [ ] Squash Tree
- [✓] Make Prototypes
- [✓] Cross-Reference
- [✓] Save Reports with Project
- [ ] Source Listing
- [ ] List Includes

Cleanscape LintPlus provides advanced test and report options that can cut hundreds of hours from code debugging and documentation for C software projects

Cleanscape LintPlus is a static source code analysis tool for ANSI C that reduces your organizational exposure to risks from latent software problems by automatically identifying problems at their source — in the C code prior to compiling or executing programs. From its first use, this venerable C source code analysis tool can save you hundreds of hours in code debugging, greatly reducing resources required for C testing efforts. Cleanscape LintPlus rigorously examines source files both individually and as a group, almost instantly generating comprehensive user-definable reports on hundreds of problems that are readily accepted by a typical compiler.

Cleanscape LintPlus is ideal for C development environments that have tight deadlines for producing high quality output, and includes enhanced support for:

- Cross-platform development
- Embedded systems development
- GNU development

This advanced source analysis tool provides improved work-flow for every step in the development process and is particularly beneficial for programmers, program managers, and QA managers. Cleanscape LintPlus also provides project managers with coding standards enforcement and statistical quality control measures by detecting noncompliance with codified style standards, by detecting maintenance or portability problems and by computing customized quantitative indicators of code size, complexity, and density.

Cleanscape Software Int'l • 5850 Eubank Blvd. NE, STE B49-180 • Albuquerque, NM 87111 • 505-246-0267 • sales@cleanscape.net
www.cleanscape.net

Cleanscape LintPlus 5

Static Source Code Analysis Tool for C

Features

Feature	Description	Command Line	GUI	OnLine
REPORTS				
Analysis	Describes problems found	■	■	■
Call Tree	Shows "calling" structure of analyzed code	■	■	■
Cross Reference	Shows a symbol table cross reference	■	■	■
Include File Tree	Generates include file trees that show nesting structure of include files used by input code	■	■	■
Prototypes	Identifies prototypes for most 'C' functions	■	■	■
TEST OPTIONS				
Check full initialization	Reports incomplete initialization of arrays or structures	■	■	■
FYIs	Generates "informational" messages in addition to warnings and error messages	■	■	■
Global mode	Checks source files as a group of problems	■	■	■
Portability	Checks for general portability issues	■	■	■
Unused locals	Report unused local variables	■	■	■
VERBOSE	Adds extended descriptions to messages	■	■	■
Strict prototypes	Adds a section to the main analysis report which lists "unprototyped" function calls	■	■	■
Unused results	Report unused function call results	■	■	■
Unusual constructs	Reports valid but unusual constructs, such as passing structures by value	■	■	■
SIGN	Control signed/unsigned checking	■	■	■
System Library	Check calls to system library routines	■	■	-
Target System	Select a target system	■	■	-
Warnings	Generates warnings	■	■	■
REPORT OPTIONS				
Call Tree	Enable/Disable Call Tree Report	■	■	■
Trim Branches	Merges redundant sub trees in Call Tree	■	■	■
Cross Reference	Enable/Disable Cross Reference Report	■	■	■
Include File Trees	Generates include file trees showing nesting structure of the "include" files	■	■	■
Make Prototypes	Generates prototypes for the 'C' functions defined in the input files	■	■	■
Pad Tree	Adds white space to call trees produced	■	■	■
Source Listing	Adds a source listing to the main analysis report; shows errors in context	■	■	■
List Includes	Generates a source listing with "include" file	■	■	■
Statistics	Adds a "statistics" section to analysis report	■	■	■
MISCELLANEOUS OPTIONS				
Linux Kernel Aware	Required for kernel aware Linux program	■	■	■
Add separators	Add separators before error messages	■	■	■
DEFINE	Define a preprocessor symbol	■	■	-
UNDEFINE	Un define a preprocessor symbol	■	■	-
DISABLE	Disable specific local error messages	■	■	-
ENABLE	Enable specific local error messages	■	■	-
STANDARD	Add standard include file directories	■	■	-
Local mode	Adds local include file directories	■	■	-
OTHER ANALYSIS OPTIONS				
ALL	Combine several options	■	-	-
ARCHAIC	Allow "archaic" initialization statements	■	-	-
BEEP	Control audible output	■	-	-
BRIEF	Skip repeated local error messages	■	-	-
FULLINIT	FULLINIT Report incomplete initialization X	■	■	■
GRAPHICS	Change the call tree graphics characters	■	-	-
HTREE	Generate "include file" trees	■	■	■
LIST	Generate source code listings	■	■	■
MAXERROR	Set max. number of local errors per module	■	-	-
MAXFATAL	Set max. number of fatal errors per run	■	-	-
MAXGSE	Enable "FYI" (informational) messages	■	-	-
NITPICK	Report valid but unusual constructs	■	■	■
PAGE	Control pagination	■	-	-
PORT	Control portability checking	■	■	■
PROTO	Generate C prototypes or FORTRAN shells	■	■	■
RESULTS	Check for unused function call results	■	■	■
SILENT	Disable progress messages	■	-	-
Snolocal	Doesn't search -I directories for "standard"	■	■	-
Snohost	Doesn't search the host system's standard-include directories	■	■	-
STRICT	Add extra checks to "global mode"	■	■	■
SYSTEM	Select a target system	■	■	-
UNUSED	Report unused local variables	■	■	-
Unused results	Report unused function call results	■	■	■
Unusual constructs	Reports valid but unusual constructs, such as passing structures by value	■	■	■
XREF	Generate a cross reference	■	■	■

Specifications

Classification

- Static Source Code Analyzer for C

Programming languages

- ANSI C

Interface

- Graphical User Interface **New!**
- Command-line interface

Development Platforms

- Microsoft Windows **New!**
  - *All 32-bit systems*
- Unix
  - *AIX: 3.X and 4.X on IBM RS/6000 based systems*
  - *Tru64 / DecUnix: Compaq/Digital UNIX 3.X and 4.X on Alpha*
  - *HP/UX: 9.X, 10.X and 11.X on HP 9000 /700 and /800*
  - *SGI IRIX: 5.3+ and 6.X on SGI*
  - *SUN: Solaris 1.X (SunOS 4.X) and Solaris 2.X (SunOS 5.X)*
- Linux
  - *Intel and Alpha systems, including: Linux, Red Hat, Debian, Etc.*

Embedded Environments

- Microtec Research ANSI C
- Plessey ARM
- Wind River Diab
- Microsoft Visual C
- Windows CE
- VxWorks
- Metrowerks
- Other generic systems
- Others added upon request

CLEANSCAPE
software international

www.cleanscape.net
505-246-0267
sales@cleanscape.net