Cleanscape C++lint User's Guide

Version 1.3.5

Sales and Service Office
5850 Eubank Blvd. NE, B49-180
Albuquerque, NM 87111
Toll-free 800-94-4LINT
505-246-0267
Fax 505-247-2678
www.cleanscape.net
sales@cleanscape.net
support@cleanscape.net

Acknowledgements

PART I Introduction

1.1 WELCOME

Thank you for your product purchase! With Cleanscape C++lint, you have the most powerful static source (lint) analysis available for C/C++ code. You also get several different user interface options, one of which, we trust, will provide the perfect level of integration for the way *you* develop software.

Cleanscape C++lint is a set of ease-of-use enhancements to the venerable PC-lint/Flexelint product from Gimpel Software (hereinafter referred to as simply PC-lint). C++lint includes a fully licensed copy. PC-lint is the longest-running tool available for C/C++ programmers, dating back to the 1980s. It is a living product with ongoing product enhancements to keep it in lock-step with the evolution of C/C++ standards.

1.2 DOCUMENTATION

This is the "quick start" guide for the Cleanscape C++lint static analyzer. There are three modes of operation:

A. *Cleanscape GUI* *

B. *Integration into IDEs using Cleanscape automation*

C. *Command line*

* **NEW** Starting with version 1.3, the GUI may be used to create the control settings for the IDE and command line modes. This is discussed in detail in Section 5 and Section 6.

This document's sole purpose is to describe the ease-of-use enhancements provided by Cleanscape Software. PC-lint is very rich in analysis controls and reporting; to gain the most benefit from your product purchase, we urge you to read and keep handy the companion document, PC-Lint Reference Manual (last bookmark in the Acrobat .pdf version of this document).

While on the topic of documentation: if you choose Cleanscape GUI or use the GUI to build the control settings for IDE or command line modes, be sure to check out the Online Help facility! It's concise yet useful information. The Table of Contents and many interrelated items in the help text are hyperlinked to make information access quick and easy.

1.3 PURPOSE

A. Function

1. Cleanscape C++lint is a programming tool that simplifies the debugging and maintenance of both large and small C and C++ programs. Cleanscape C++lint provides ease-of-use enhancements to, and a fully licensed copy of, the venerable PC-lint/Flexelint product from Gimpel Software.
2. The PC-lint source code analyzer that can detect over 1000 potential problems, including:
   a. Inappropriate arguments passed to functions

b. Inappropriate library calls
c. Non-portable code
d. Type usage conflicts across different modules
e. Unused variables and dead code

B. Application

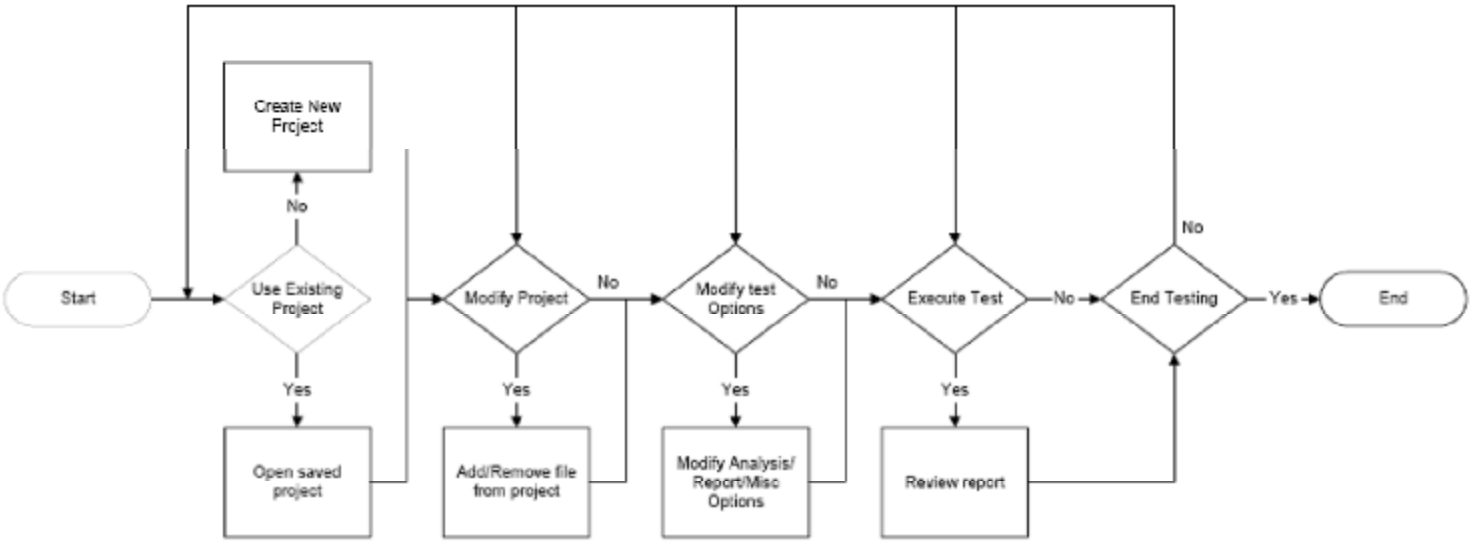
1. Cleanscape C++lint can be used to:
   a. Check source files before they are compiled
   b. Isolate obscure problems
   c. Identify problems before debugging is required

C. Advantages

1. The diagnostic messages produced by Cleanscape C++lint are more detailed than those produced by standard compilers, and cover a wider range of syntactic and semantic problems.
2. Cleanscape C++lint analyzes source files both individually and as a group, and can therefore identify problems that are beyond the scope of a compiler.
3. Cleanscape C++lint is effective in reducing development time and improves C/C++ programming style.

D. Flow of Analysis

1. The following flowchart illustrates the Cleanscape C++lint test process:



PART II Requirements, Installation, and Uninstallation

2.1 WINDOWS

A. System Requirements

1. Hardware
   Any configuration sufficient to run Windows is sufficient for C++lint.
2. Operating System
   a. Microsoft Windows 98® and 98® SE
   b. Microsoft Windows NT® 4.0 with Service Pack 6a (SP6a)
   c. Microsoft Windows 2000® with Service Pack 2 (SP2)
   d. Microsoft Windows XP® with Service Pack 2 (SP2)
   e. Microsoft Windows Vista®
3. Web Browsers
   a. Firefox® 1.2 or above
   b. Microsoft Internet Explorer® 5.x or above
   c. Mozilla® 1.7 or Netscape Navigator® 4.7x or above
   d. Opera® 6.x or above

B. Software Setup Procedure

1. Installation

   a) Download `cpplintgui<ver>_win.exe` to a temporary directory, then run it.

   b) An installer window will appear and should extract a number of files to the installation directory you specify (hereinafter referred to as `<install_dir>`; the default is `c:\cleanscape\cpplint`). The installer exits automatically, and no reboot is required.

   c) The installer automatically creates a shortcut for the Cleanscape C++lint GUI on the desktop. To run the GUI, double-click the shortcut follow the instructions to obtain a license key as described in Section 3.

   d) The installer searches for any supported versions of Visual Studio on your machine and if found, automatically installs the Cleanscape macros. NOTE: This does not apply to Windows 98! If you are a Win98 IDE user, please contact support@cleanscape.net.

   e) Finally, the installer adds the "main" subdirectory to your system PATH – necessary for running C++lint (or any of its associated support programs) from the command line. To do this manually enter the following command: `set PATH=<install_dir>/main;%PATH%`

2. Additional steps for Windows 2000

   If you're going to install Cleanscape C++lint under Windows 2000 as Administrator, and you want to make the program accessible to ordinary "Users", some additional steps are required. For more information, see Section 7.1.

C. Uninstallation

1. Using the included uninstaller

   There is an uninstaller named "uncpp.exe" in the "main" subdirectory. To run, double-click on it or run it by name in a command prompt. By default (with no operands), the uninstaller will remove all components as described in step 2 below. If desired, it's also possible to delete each component on a piece-by-piece basis. Run any of the following from a command prompt:

   - `uncpp -ICON` Remove the C++lint GUI icon from the Desktop.
   - `uncpp -PATH` Remove Cleanscape C++lint from the system PATH.
   - `uncpp –VS [`*`ver`*`]` Uninstall the Cleanscape macros from Visual Studio on your system. NOTE: Not available for Win98.
   - `uncpp -DIRS` Queues <*`install_dir`*> for deletion upon next login.

2. Manual installation – Section d) required for Windows 98

   a) Delete the installation directory and its subdirectories.

   b) Remove the Cleanscape C++lint shortcut from the desktop

   c) Delete the Cleanscape directory from your PATH:
   - In Windows 98, delete the appropriate "`set path=`" statement from your `c:\autoexec.bat` file.
   - In Windows NT/2K/XP/Vista, right click your "My Computer" icon on the desktop, select "Properties", click the "Advanced" tab, click the "Environment Variables" button, double-click the text field "Path" in the System Variables area, and from that string, delete <*`install_dir`*>`\main`

   d) If Microsoft Visual Studio is installed on your machine, macros were automatically integrated upon C++lint installation (see Section 5). Delete these macros using the following steps:
   - Open your Visual Studio IDE.
   - Select the Tools dropdown menu.
   - Select "External Tools…" (for VS 6, select "Customize…", then click the "Tools" tab in the dialog).
   - Click on each Cleanscape macro in turn, then click on the Delete button (for VS 6, delete each macro by clicking the red 'X' in the top right).

3. Restore your system to the point just before Cleanscape C++lint installation – not available for Windows NT/2k

   The installer created a system restore point just prior to installation. If you have not added new programs in the interim, you can safely roll your system back to this point. For Win98, use `scanreg /restore`

2.2 UNIX/LINUX

A. System Requirements

1. Hardware
   A minimum of 256 MB memory is required for Cleanscape C++lint.
2. Operating System. Note the GUI version may differ amongst the various hosts.
   a. Most GNU/Linux OSes, including RedHat®, SuSE®, Debian®, Ubuntu®
   b. HP HP-UX®
   c. IBM AIX®
   d. SGI Irix®
   e. Sun Solaris®
   f. Mac OS-X® Tiger
3. Web Browsers
   a. Firefox® 1.2 or above
   b. Mozilla® 1.7 or Netscape Navigator® 4.7x or above
   c. Opera® 6.x or above

B. Software Setup Procedure

Installation – installation as root is easier and recommended. Refer to the installation notes for details. The '#' below represents the root prompt.

a) Download the latest version of `cpplintgui<ver>_<OS>.taz` to a temporary directory, e.g., `/tmp`.

b) Create installation directory, e.g., `/usr/local/cleanscape`, and `cd` to it.
NOTE: On Unix hosts, set environment variable `CSIAPPBASE` to this directory.

c) Use the following commands to extract the files:
```
# gunzip  /tmp/cpplintgui<ver>_<OS>.taz
# tar xpvf /tmp/cpplintgui<ver>_<OS>.tar
```

d) Start the GUI:
```
# cpplintgui &
```
and follow the instructions to obtain a license key as described in Section 3.

e) If this is a server-based application, start the daemon on the server as root:
```
# startup
```
NOTE: The daemon must be running on the server before clients can access/ use the product.

f) If you intend to run Cleanscape C++lint from the command line, two additional commands are required (examples below are for sh/bash):
```
# export IPTLINT=<install_dir>/cpplintgui.dir/main
# export PATH=$IPTLINT:$PATH
```

C. Uninstallation – manual process

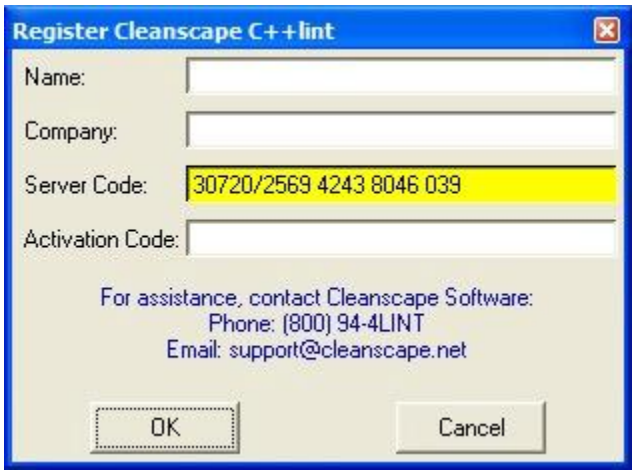
a) Delete the installation directory and its subdirectories.
b) Delete files `myeditor.lst` and `ctemplate.csi` from all users' `$HOME` directories.

PART III Activation; Initial Test

A. Registration Process – GUI

The first time that you run the program, a registration prompt will be displayed. You must "register" the program before you can use it. This will run again if the activation key (provided by Cleanscape) expires.

1. If Cleanscape C++lint is not registered, a notification will be displayed. To continue, press OK. The following dialog will appear (NOTE: your server code will be different!):



2. To obtain your activation key, contact Cleanscape Software and provide the server code listed on the "Register Cleanscape C++lint" window. The "Server Code" is on the line with a yellow background. HINT: Highlight the server code with your mouse. Unix users, you now have a copy; Windows users, type CTRL-C on your keyboard to copy the server code.

   To reach Cleanscape, call 800-94-4LINT or +505-246-0267, or send email to support@cleanscape.net.

3. Enter your name, company name and the activation key. HINT: Copy the activation key from your email from Cleanscape, then paste it in the activation code text box (Windows users, type CTRL-V on your keyboard). When ready, press OK.

4. You should then see an "About Cleanscape C++lint" dialog box. At this stage, Cleanscape C++lint is now registered and operational. If you do not see this dialog box, contact Cleanscape for further assistance (see #2 directly above for contact info).

B. Registration Process – IDE integration and command line

0. If you wish, you can register the product using the GUI – the license key is created and stored correctly using either method. Just be sure to set up the environment variables per the instructions in Section 2.1.B.1.e or 2.2.B.f above! If you use the GUI method, you can skip the rest of this section.
1. Be sure you have set up the environment variables per the instructions in Section 2.1.B.1.e or 2.2.B.f above.
2. Run the command, `cpplint -license=activate`

   Hit <Enter> to leave the number of license servers at its default of 1.

   The next line from the activation program will contain your server code. On Windows machines, it starts with "30720/"; on Unix/Linux, it is purely numeric.

   To obtain your activation key, contact Cleanscape Software and provide this server code.

   To reach Cleanscape, call 800-94-4LINT or +505-246-0267, or send email to support@cleanscape.net.
3. Once the activation key is entered, Cleanscape C++lint is registered and operational.

C. Initial Test

Once registered, you need to verify that the system ("library") include directories are properly identified and located by C++lint. Most Windows compilers use the INCLUDE environment variable, which C++lint reads automatically; verify that the order within INCLUDE is correct, or override as described below.

If you have any difficulties with this step, email support@cleanscape.net

GUI

1. Select your target compiler, e.g., `AIX` if running a native AIX host, `gcc_latest` for Linux or compiling with gcc, or other native or embedded processor. Select `Errors_only` in the Warning Level dropdown.
2. Click the Add Files button. In the resulting dialog box, navigate to the examples subdirectory, then select `testcfg.c` or `testcfg.cpp`, depending on whether you are building C or C++ applications.
3. Click the Run button.

   If you selected `gcc_latest` or `g++`, command line program setupgcc will run to determine site-specific internal settings such as defines, macros, and include directories.

   Check the results in the Reports window (right side of GUI). If there are no errors, you are ready to run; if there are errors, add the system include directories separated by spaces (e.g., `/usr/include`) to the Include Directories text box on the Misc Options tab in the lower left frame of the GUI.

Command Line

1. If you intend to gcc or g++, run command line program setupgcc first to determine internal settings such as defines, macros, and include directories.
2. Specify your target compiler by adding the appropriate `co-`*`xxx`*`.lnt` file from the `main/lbin/lf` subdirectory.
3. Test your command line against `examples/testcfg.c` or `testcfg.cpp`

   Observe the results produced. If there are no errors, you are ready to run; if there are errors, add the system include directories (e.g., `-i/usr/include`) to the command line.

The next three sections describe in detail the operation of Cleanscape C++lint

- from the GUI Part IV
- integrated within IDEs Part V
- from the command line Part VI

PART IV Running the Cleanscape C++lint GUI

A. Overview

The Cleanscape GUI is a tried-and-true graphical interface used successfully for years on Cleanscape's Fortran-lint and Lint-Plus for C static source analyzers. It is also the planned interface for future C/C++ and Java analyzers and test tools.

The Cleanscape GUI is ideal if your primary development environment centers on a programming editor (as opposed to an IDE, as we'll discuss in the next section). The Cleanscape GUI provides hyperlinking between the results (in the Reports frame) and the line of source in the source file that caused the message.

Advantages of the Cleanscape GUI include:

- Fast
- Easy to learn, navigate, and use
- Information readily at the programmer's fingertips
- Point-and-click control for options-laden PC-lint command-line product. NOTE: Your suggestions to improve this ease-of-use feature are appreciated! Email suggestions/comments to sales@cleanscape.net.

NEW Starting with version 1.3, the GUI may be used to create the control settings for the IDE and command line modes. This is discussed in detail in Section 5 and Section 6.

Supported code editors are listed below. It is also possible for users to integrate their own editor; see Section 7.2 for details on the seteditor program. For an updated list, visit http://www.cleanscape.net/products/cpp/IDE_Ed.html.

Windows editors:

- Borland CodeWright
- Crimson Editor
- Emacs
- Epsilon Programmer's Editor
- Gvim
- GWD Text Editor
- MultiEdit
- Starbase CodeWright
- TextPad
- UltraEdit
- Visual SlickEdit
- Visual Studio 6 *
- Visual Studio .NET 2003 *
- Visual Studio 2005 *
- Visual Studio 2008 *
- Visual C++ Express 2008 *

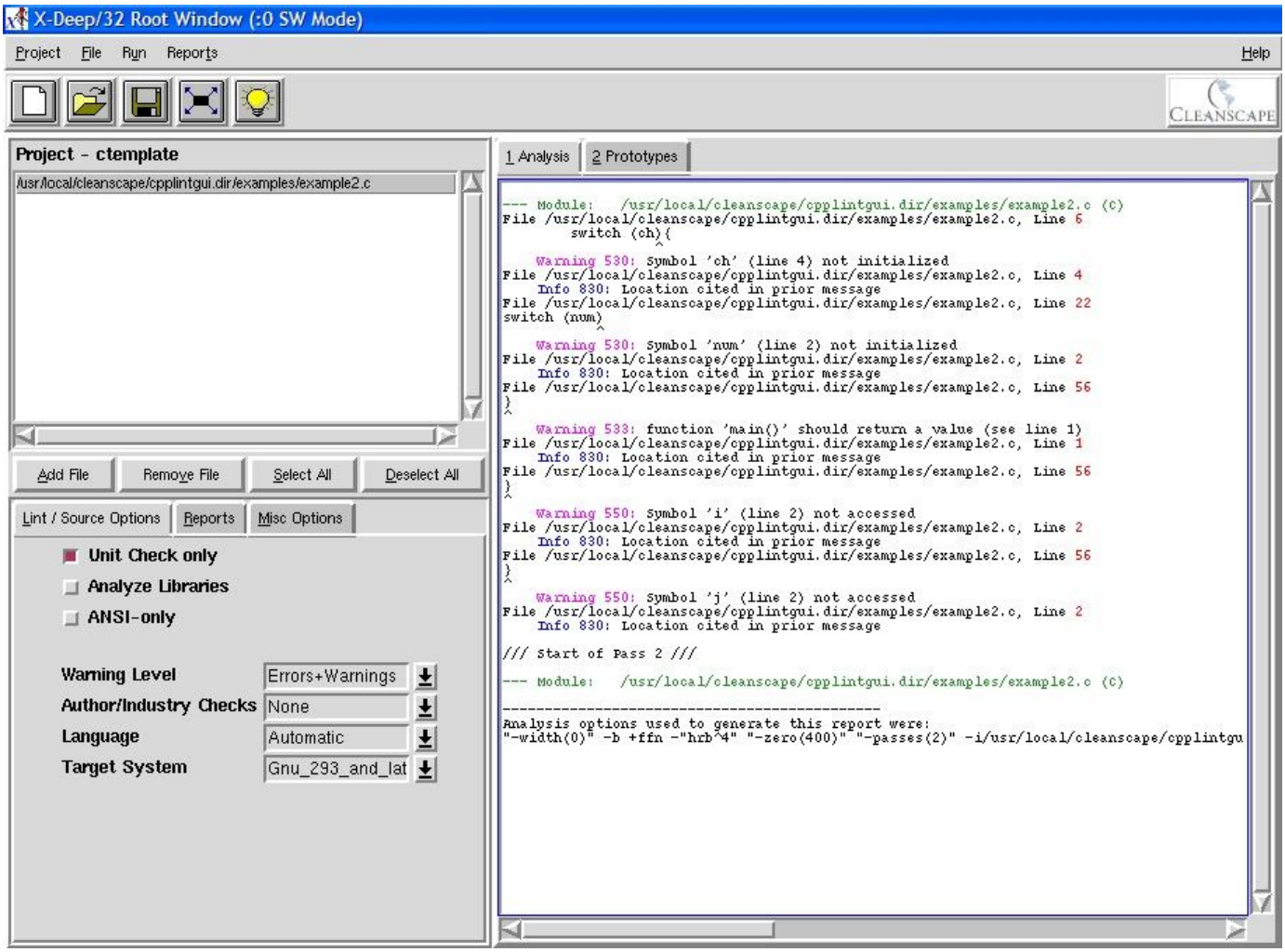
Unix/Linux editors:

- Elvis
- Emacs
- Jed *
- Joe *
- Nano *
- Nedit
- Pico *
- Vi *
- Vim *
- XEmacs

* Multiple instances of these editors will open with each link click.

All elements of the GUI are also controllable from the keyboard; this is discussed in Section K below.

> **NOTE:** A sample Unix/Linux screenshot is shown below. All subsequent screen shots are Windows-based, but the functionality is identical between the two environments.



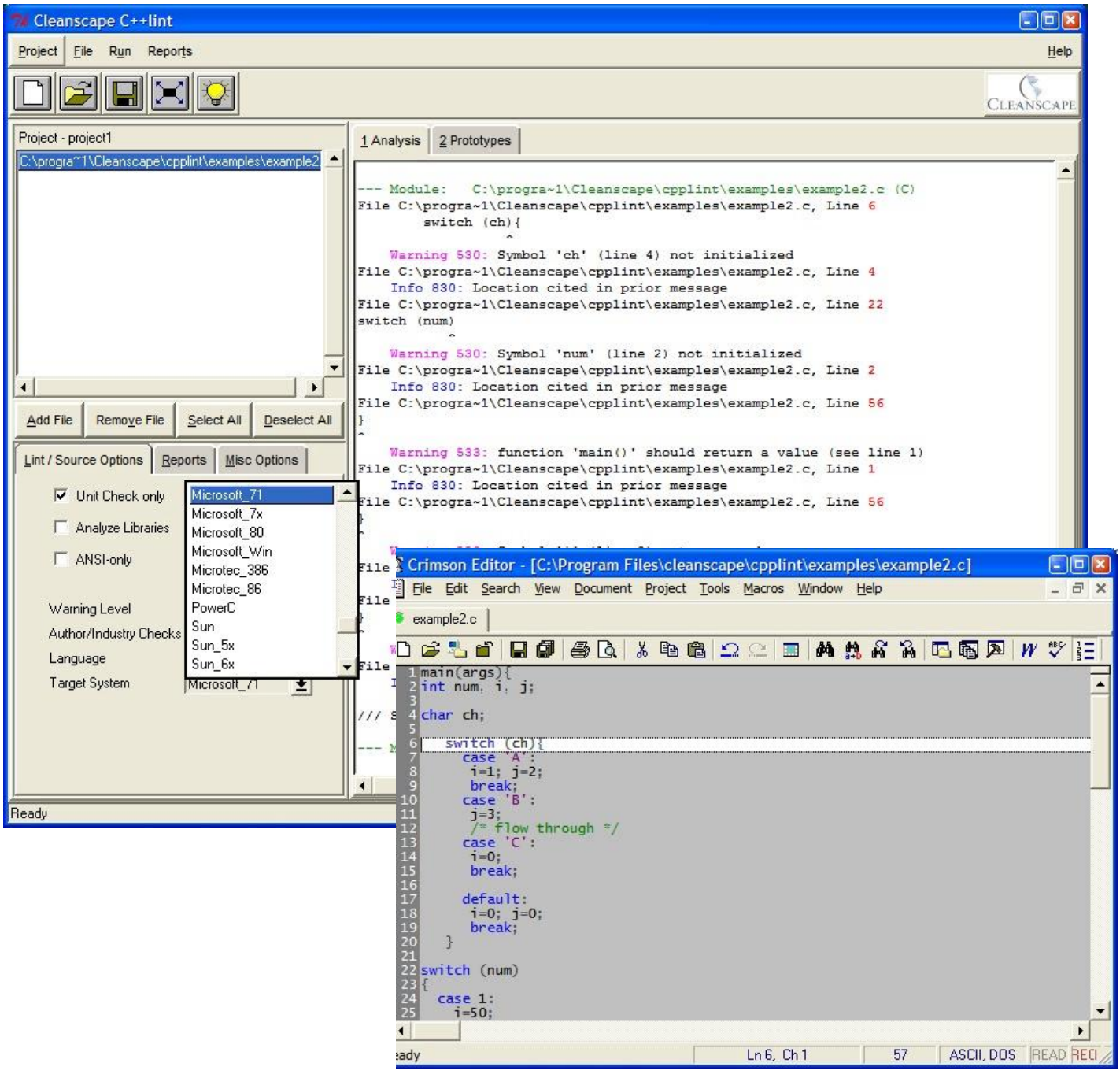
The screenshots above and below depict a sample Cleanscape C++lint session. The "Target System" dropdown makes it easy to select from over 90 supported compilers – native as well as embedded. On the next page, the lower screenshot is of Crimson Editor (previously selected as the External Editor in the Reports tab in the lower left frame), activated when the red "6" hyperlink near the top of the Analysis Report was left-mouse-clicked. C++lint positioned the editor to the line in the source file that caused the analysis result. Messages numbers are color-coded according to their severity; "Warning 530" is an example.

The hyperlink feature also makes it very easy to add PC-lint "magic comments" to modify analysis at the statement or block level. For more information, see Sections 5.1 and 5.2 of the PC-Lint Reference Manual (last bookmark in the Acrobat .pdf version of this document).

Also, it is possible to open any file listed in the Project window (upper left frame of the GUI) by right-mouse-clicking on the desired filename.

The Cleanscape C++lint GUI also remembers settings (e.g., checkboxes, target system, external editor) from the previous session by creating a template file named `template.csi` in the `main` subdirectory or your `$HOME` directory if Unix/Linux. There is no template file upon installation.

Finally, note that it is possible to add compilers to the Target System dropdown by editing file `<install_dir>/bin/mycompiler.lst`; in fact, that is how Cleanscape will add newly supported compilers. See the comments in that file for details.



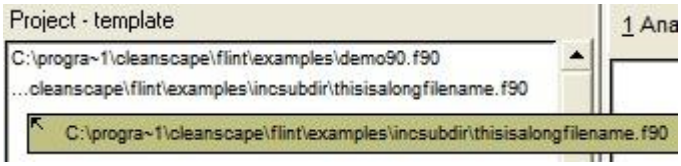
B. Components

Where possible, each component features "balloon" help which will appear if you hover the mouse over an item or control description. Additional help for each item may be found in the Online Help (see Section 4.I).

1. Program menu: 

2. Shortcut bar: 

3. Project window:

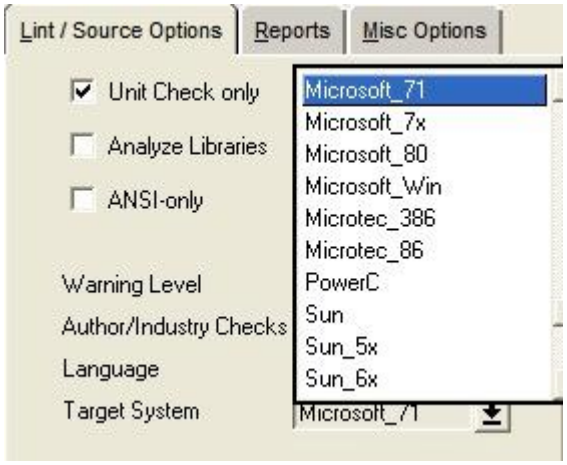


Any file listed in the Project window can be opened in the selected editor by right-mouse-clicking the filename. Any filenames too long to fit the window are shorted to ~60 characters and an ellipsis is prepended. The full filename appears in a balloon tip if hovering the mouse over the name, as shown above.

4. Project shortcut buttons: 

5. Lint/Source options window (with "Target System" dropdown activated). Currently, Cleanscape C++lint supports over 90 compilers/versions, including many embedded targets. This point-and-click control makes using the options-laden PC-lint command-line product easy! NOTE: Your suggestions to improve this feature are appreciated – email them to sales@cleanscape.net.

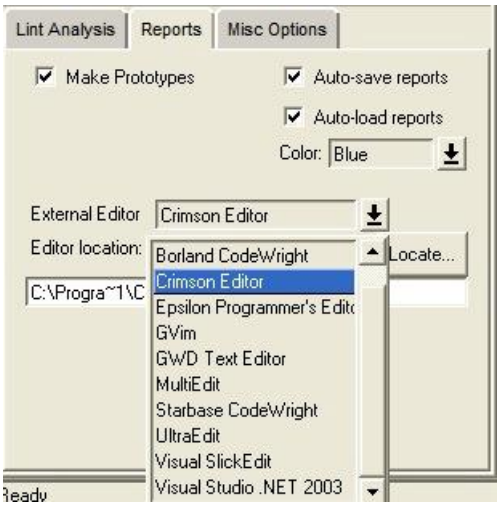
USAGE NOTE 1: Due to the rich extent of PC-lint analyses, it is recommended that first-time projects turn off all Warnings ("Errors_Only" selected), then progressively add levels of warnings once prior analyses are assessed/addressed. Individual analyses can be enabled/disabled by number in the appropriate text box on the Miscellaneous Options tab, discussed below.



USAGE NOTE 2: If you specify `Gcc_Latest` or `g++` as the Target System, its first use will cause an external program, `setupgcc`, to run. Follow the dialog

box and onscreen instructions to automatically gather information about your specific gcc/g++ build.

6. Report Options window (with External Editor dropdown activated):

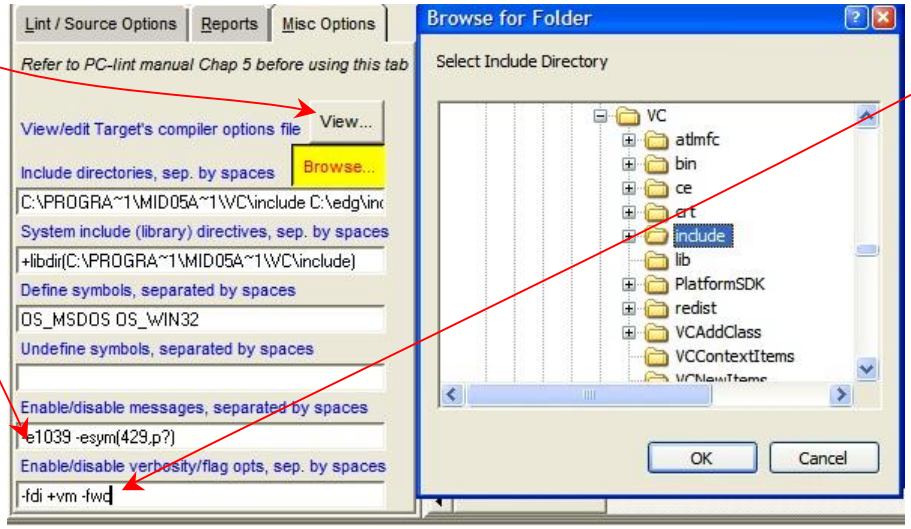


7. Miscellaneous Options window (with "Browse Includes" feature activated).

   USAGE NOTE 1: Starting with version 1.3.5, one can now view/edit the co-*xxx*.lnt file associated with the Target compiler by clicking the View… button. Default actions for your compiler are set there and can be modified.

   USAGE NOTE 2: Cleanscape C++lint also adds various control mechanisms based on the type of source files (e.g., .c vs. .cpp) and the Language option from the Lint/Source Options tab. View these controls at the bottom of the Analysis report, immediately after the line, "Analysis options used to generate this report were:". Override as needed by adding –<*opt*> in the last text box.

   USAGE NOTE 3: Individual analyses can be enabled/disabled by number in the "Enable/Disable messages" text box (e.g., +e710 –e844 –e322). For more information on the various settings that can be controlled from this tab, see Sections 5 of the PC-Lint Reference Manual. For a complete list of analyses by number, see http://www.cleanscape.net/products/cpplchecks.html



8. Report windows:

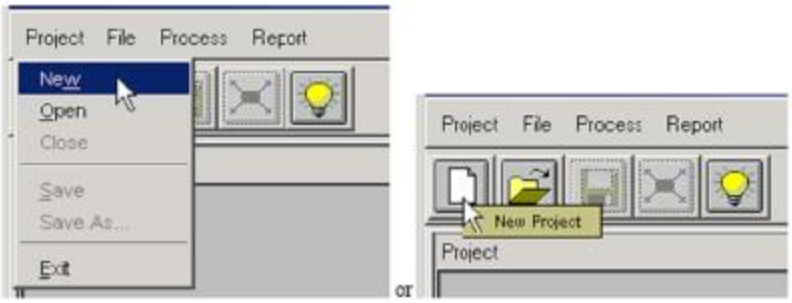
   Example reports 

   appear in Section H below.

C. Creating a new project

0. In Cleanscape terminology, a *project* is a collection of file *names*, include directories, define/undefine definitions, and analysis controls. Actual files are not relocated on the disk, nor are they deleted if removed from a project.

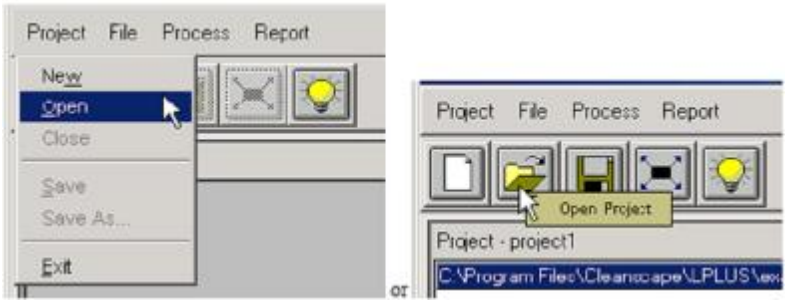
1. To create a new project, select Project/New from the menu or press the New Project button on the shortcut bar. Note: If a project is already open, a dialog box will prompt you to save the old project first.



2. A new project name appears in the title, but can be renamed upon a save.

D. Opening an existing project

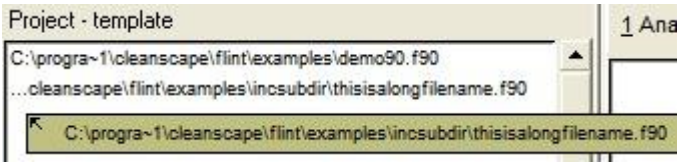
1. To open an existing Cleanscape GUI project, select Project/Open from the menu or press the Open Project button on the shortcut bar:



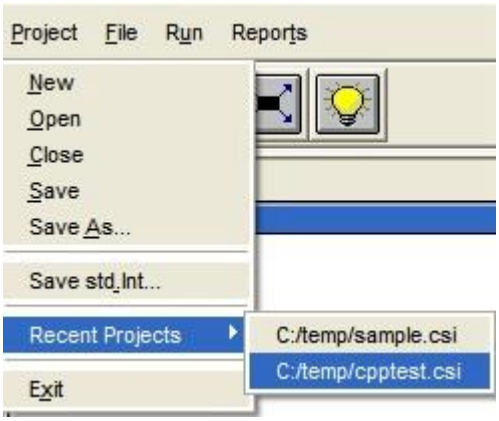
2. A standard Open dialog box will appear:

a. Browse to find/select a project file (with extension `.csi`).
b. When ready, press the Open button in the lower right corner.

3. Files associated with the project are displayed in the file listing window:

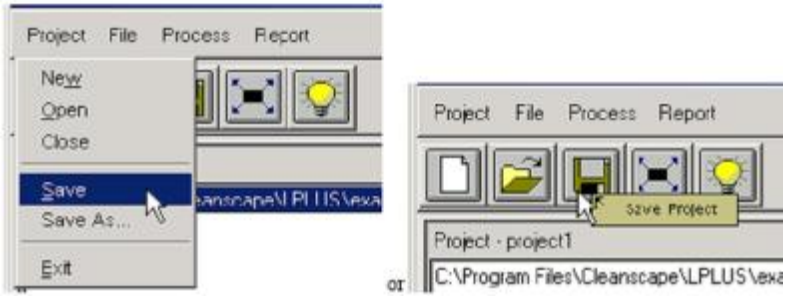


4. It is also possible to open recent projects using the Recent Projects menu:



E. Saving a project

1. To save the current state of a project, select Project/Save from the menu or press the Save Project button on the shortcut bar:



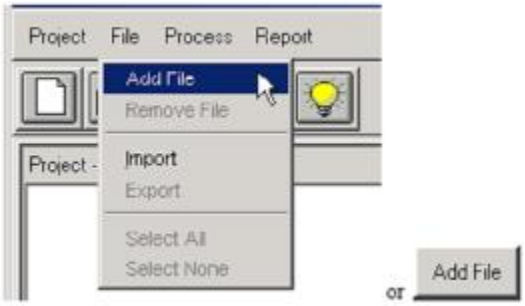
2. If this is a new project, the Save As window will appear.
   a. Enter a name for the project.
   b. When done, press the Save button.
   c. You can also use the "Save As..." feature in the Project dropdown to save an existing project under a new name.

3. **NEW** Starting with version 1.3, the GUI may be used to create the control settings for the IDE and command line modes. This is discussed in detail in Section 5 and Section 6. Use the "Save std.lnt" option as shown:

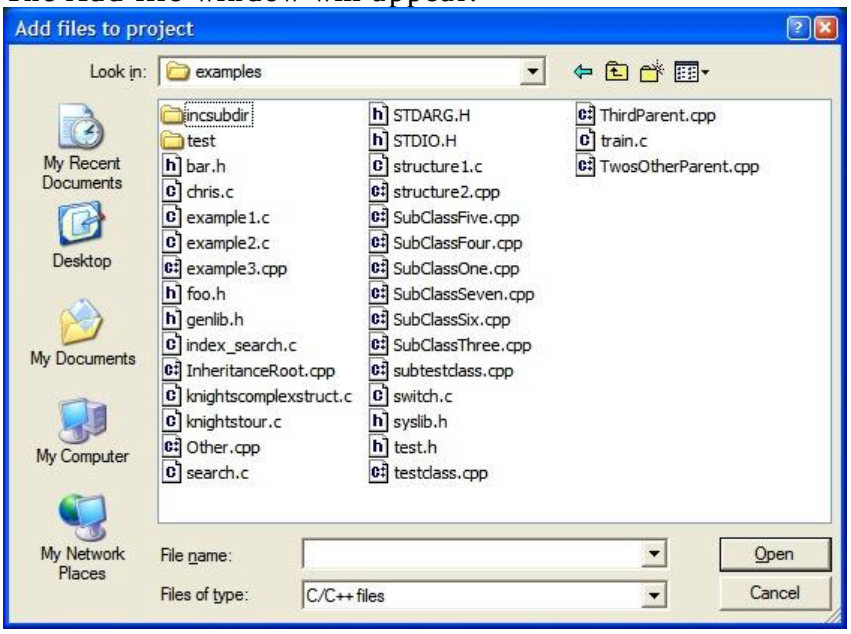
F. Modifying a project

1. Add files to a project

   a. To add one or more files to a project, select File/Add File from the menu to add files into the project or press the Add File button on the project shortcut bar:



   b. The Add file window will appear:



   c. C/C++ source files are the default file type.

   UNIX NOTE: The default file type is .c, which can be modified by entering the appropriate type (e.g., *.cpp) in the Filter textbox at the bottom of the dialog. It is also possible to permanently modify the filter type by editing the "Default Add File filter" line in text file `~/cpplint.ini`.

   1) The file-selection dialog supports multiple-file selection under both MS-Windows and UNIX.

   2) To add multiple files individually, use <Control> + Left Mouse Button. Each selected file will be highlighted.

   3) To add a group of files:
      (i) Left-click on the first file.
      (ii) Hold down the <Shift> key.

(iii) Click the last file. The first, last, and all in-between will be highlighted.
(iv) When done, press the Save button.

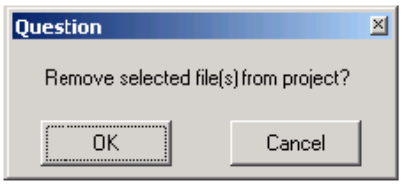
2. Selecting files for the analysis run

Files within the project may be selected/deselected before an analysis is run. To do so, use the same highlighting techniques as described just above. To select or deselect all files, use these buttons: 

3. Removing files from a project

a. To remove individual source files from a project, select the files to be removed (as explained in the preceding section) and then press the Remove File button. To remove all files from a project (i.e., to clear the file list), first press the Select All button, and then press the Remove File button.

b. Press the OK button to confirm the removal operation:



c. The updated file list is displayed in the project window.

d. Note that this operation has no effect on the actual file on-disk.

G. Execute Analysis

1. Create a new project or open an existing project.

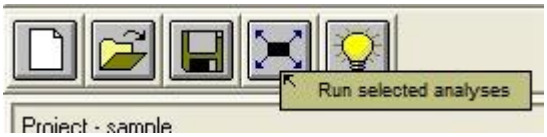
To create a new project, see Section 4.C.
To open an existing project, Section 4.D.

2. Select the files to be analyzed as explained in Section 4.F.1.

3. Modify options as necessary, using the tabs in the lower left frame of the GUI, as displayed in Sections 4.B.5-7. See balloon help, Online Help, and the PC-Lint Reference Manual (last bookmark in the Acrobat .pdf version of this file) for descriptions of each option.

4. To analyze the selected files, use Run/Go from the menu or press the Execute test button on the shortcut bar:

or



H. Review reports

1. To view the generated reports, click on the appropriate report tab:

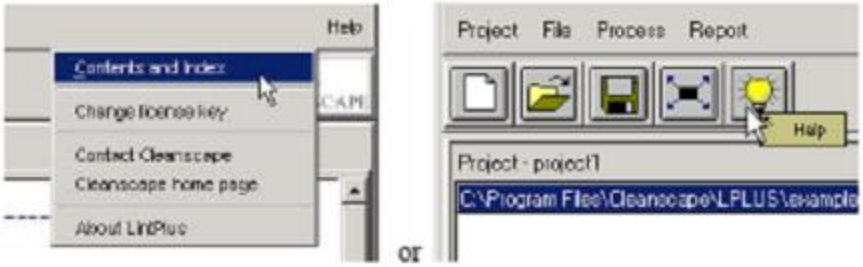
2. To print reports, or to save them to disk, use the Report menu dropdown at the top of the screen. Reports may be printed or saved collectively or individually.
3. Samples of each of the reports are depicted below. Remember that clicking any entry in red will open the source file at the appropriate source line in the specified External Editor. Sample Analysis and Generated Prototypes Reports:

I. Online Help

The Online Help System contains concise yet useful information for running the Cleanscape GUI. The Table of Contents and many interrelated items in the help text are hyperlinked to make information access quick and easy.

1. Accessing the Help System

   a. To access the online help system, select Help/Contents and Index from the menu or press the Help button:

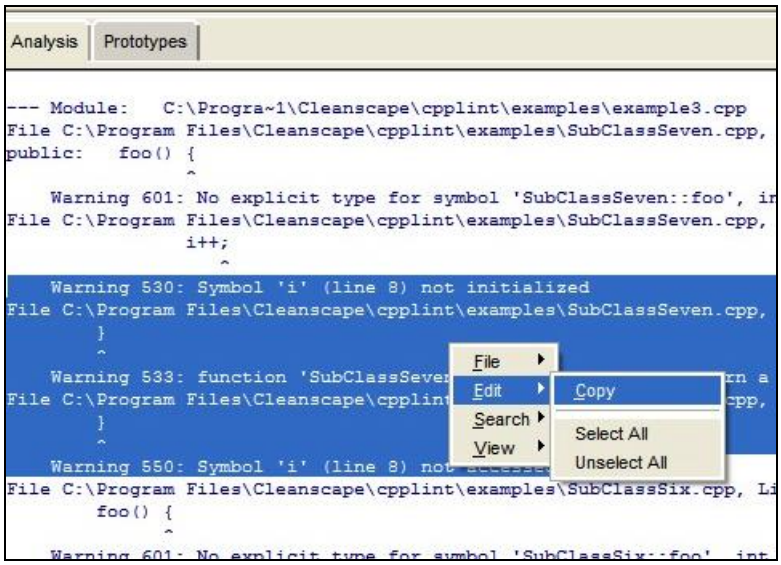


   b. The Cleanscape C++lint Help browser will appear:

J. Sub-Menu Functions

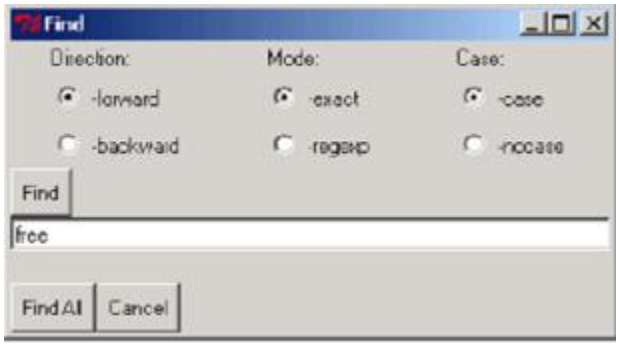
There are several "right-mouse-click" options available while in the Reports frame on the right hand side of the GUI. These features should be self-explanatory for those familiar with graphical environments. The more commonly used features are shown in detail below.

1. Copy
   a. Press the right mouse button inside reports frame
   b. Select Edit -> Copy



   c. The text can now be pasted into other applications (e.g., Microsoft Word).

2. Search
   a. Press the right mouse button inside a report frame.
   b. Select Search -> Find.
   c. Enter string to search and select the desired options:



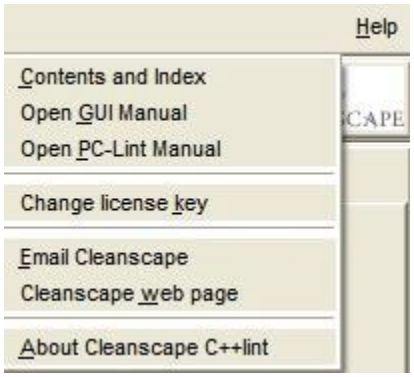
   d. The search result(s) will be highlighted.

K. Operating the GUI using the Keyboard; Keyboard Shortcuts

All aspects of the Cleanscape C++lint GUI can be controlled from the keyboard. This capability was added to comply with the US Government's Section 508 provisions.

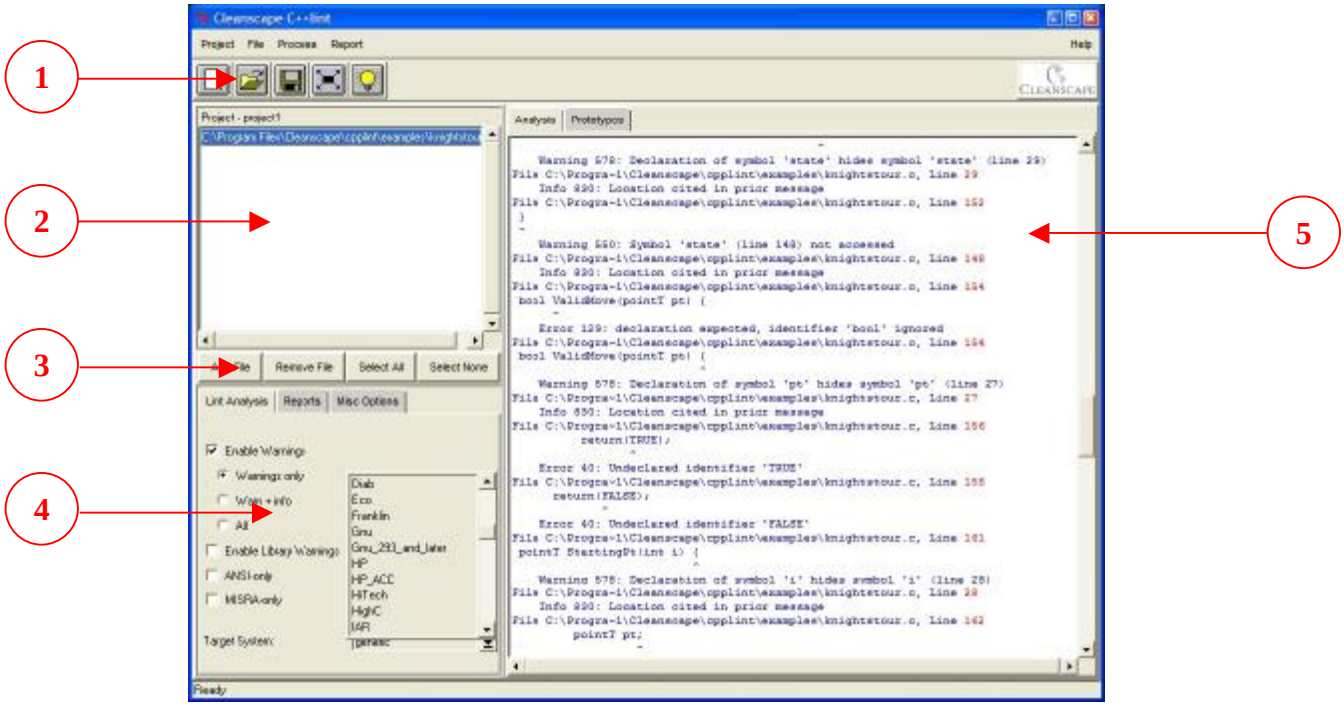
1. Accessing dropdown menus and items using keyboard accelerators. This is the standard mode common to all Windows products.

   a. Select the desired menu by holding down the <ALT> key, then pressing the underlined letter for that menu item. For instance, this screen image was obtained by pressing and holding <ALT>, then typing the "h" key:



   b. To open the GUI manual, release the <ALT> key and press "g".

2. Navigating amongst screen elements. There are 5 screen elements in the GUI, as shown below:



   a. The <TAB> key scrolls between these five screen elements and all active items within each element. <SHIFT>+<TAB> reverses the scrolling. The item with focus will have a dotted line around its border. *Note:* Because

of the background color, the icon buttons in Element 1 will not show the dotted-line highlighting.

b. For buttons (including radio buttons), pressing the space bar will "push" the button.

c. For checkboxes, pressing the space bar will "check/uncheck" the box.

d. For dropdown boxes, pressing the space bar will open the dropdown; the up/down arrows will navigate the dropdown, and the <ENTER> key will select.

3. Keyboard shortcuts.

a. The standard Windows shortcuts are available. For instance, pressing <F1> will bring up the Help listing; <ALT>+<F4> exits the program.

b. Use the alt-key combination to access a menu, then type just the underlined letter to access a submenu item. For instance, to invoke Project-Save As, one would type <ALT>+<p>, then <a>. Alternately, the arrow keys can be used to navigate submenu selections once the menu dropdown has been activated with <ALT>+<p>.

c. The following keyboard shortcuts are also available within the GUI:

<ALT>+<o>	Open Project
<ALT>+<g>	Run the Analysis (Go)
<ALT>+<x>	Exit GUI
<ALT>+<l>	Jump to Lint analysis tab (in Element 4)
<ALT>+<r>	Jump to Reports tab (in Element 4)
<ALT>+<m>	Jump to Misc Options tab (in Element 4)
<ALT>+<1>	Jump to Report #1 (Analysis report in Element 5)
<ALT>+<2>	Jump to Report #2 (Prototype report in Element 5)

L. Changing fonts / sizes

To change the fonts and sizes within the GUI, use a text file to edit the `cpplint.ini` file located in the `main` subdirectory on Windows or your `$HOME` directory if Unix/Linux.

In that file, you will see a section starting with `[fonts]`. Change the values from `default` to a value specified as follows:

```
name size style
```

where `name` is any font name on your system defined by a single word;
`size` is an integer font size, and
`style` is one of: `normal` `bold` `italic` `underline`

NOTE: The GUI makes no attempt to validate the font, size, or style. Windows are not resized due to such changes, and results are not defined if the specified font is not valid on your system.

PART V Running Cleanscape C++lint from IDEs

A. Overview

For customers who develop their code in Integrated Development Environments (IDEs), it is more useful to operate C++lint from within the environment. Advantages include:

- Cleanscape C++lint reads project information (especially the file list, include directories, and definitions) – no need to re-specify.
- Deep integration: C++lint uses the output window of the IDE for results and hyperlinks to the source files using the IDE's internal editor.
- Fewer windows to shuffle between.

NOTE: At present, only Microsoft Visual Studio is supported: VS6, VS2003.NET, VS2005, VCX2008, and VS2008. Other IDEs may be supported via the External Editor option in the Cleanscape GUI (see Section 4.B.6). Eclipse integration is underway on both Unix and Windows. To request deep integration for your IDE, email sales@cleanscape.net.

B. Installation

Important note: The installers will not operate in Windows 98! If you are a Win98 IDE user, please contact support@cleanscape.net.

To run C++lint from IDEs, you need to have registered the product as described in Section 3.

Installation for all Visual Studio versions occurred automatically when you installed Cleanscape C++lint. If you added a new version of Visual Studio or otherwise want to rerun an IDE installer, please read on.

Available installers are in directory `<install_dir>\ideinstall`

Select the installer corresponding to the version of the IDE you wish to integrate with. Other IDEs will be supported in the future; to request deep integration with your IDE, email sales@cleanscape.net.

From the ideinstall directory, type the name of the installer you wish to run. After a few moments, you will see

```
The operation completed successfully
```

in the command window. This indicates that the Cleanscape macros have installed correctly and that Cleanscape C++lint is available directly within your IDE! If you encounter any problems, please email support@cleanscape.net.

C. Operation

For Microsoft Visual Studio, the installer placed five "External Tools", found in the "Tools" dropdown menu. A description of each tool follows. In the titles for each tool, below as well as on the "Tools" dropdown menu, the underlined letter indicates the default "accelerator key" for that tool.

Clicking ALT-T then the underlined accelerator key represents a shortcut to running that tool. For details on the actual operation of PC-lint and its control and reporting options, refer to the companion document, PC-Lint Reference Manual (last bookmark in the Acrobat .pdf version of this file).

1. PC-lint Single File - macro by Cleanscape

   Select this command to run Gimpel PC-lint over the current file. Analysis results appear in the Output window. Double-clicking on any error message will cause Visual Studio's built-in editor to jump to the line number in the source file causing the analysis message.

   This is a "standalone" mode and can be applied to any single file.

2. PC-lint Project Setup - macro by Cleanscape

   NOTE: This option must be run first to successfully analyze projects or modules with macros 4 or 5.

   This option does not run any analyses, but instead parses the Visual Studio project file for the list of source files, include directories, and defines comprising the current project. These values are then placed in a PC-lint `<projectname>.lnt` file, created and maintained by this macro.

   Rerun this macro each time the project changes (changes to the source file list, include directories, or defines). Because the `<projectname>.lnt` file will be rewritten each time the macro is run, do not edit the file!

   Separate from `<projectname>.lnt`, a `std.lnt` file is used to specify analysis control settings to PC-lint. `std.lnt` is never changed automatically, and is therefore the place to set PC-lint options on a per-project basis.

NEW Starting with version 1.3, you can use the Cleanscape GUI to generate analysis switches for you. To do so, set the analysis options within the GUI as described in Sections 4.B.5-7 and the Online Help. When done, select File – Save std.lnt as shown at right; `std.lnt` will be saved to your project's root directory.

The Project Setup macro will ask you if you wish to use the Cleanscape GUI in this manner. If the GUI is not used and no `std.lnt` exists already, Project Setup will create a skeleton `std.lnt` file in the project's root directory.

Some usage notes:

1. The GUI's "Target System" setting is saved within a comment – for IDEs, the compiler value is set automatically by the Cleanscape macros. Do not uncomment this line if using an IDE.
2. If `std.lnt` already exists, it will not be altered.
3. Files are automatically determined when running the Visual Studio Project Setup macro; any GUI entries are ignored.

After creation, this macro opens `std.lnt` in Notepad. This provides your first chance to manually enter project-specific control settings for PC-Lint. Read the comments in the `std.lnt` file for startup information.

Rerun this tool each time the project contents change (e.g., when files are added or deleted). If a `std.lnt` file exists, its contents and any changes made to it are preserved.

3. PC-lint Edit LNT File - macro by Cleanscape

   This macro conveniently invokes Notepad to edit the "`std.lnt`" file (described in #2 above) associated with the current project.

4. PC-lint Entire Project - macro by Cleanscape

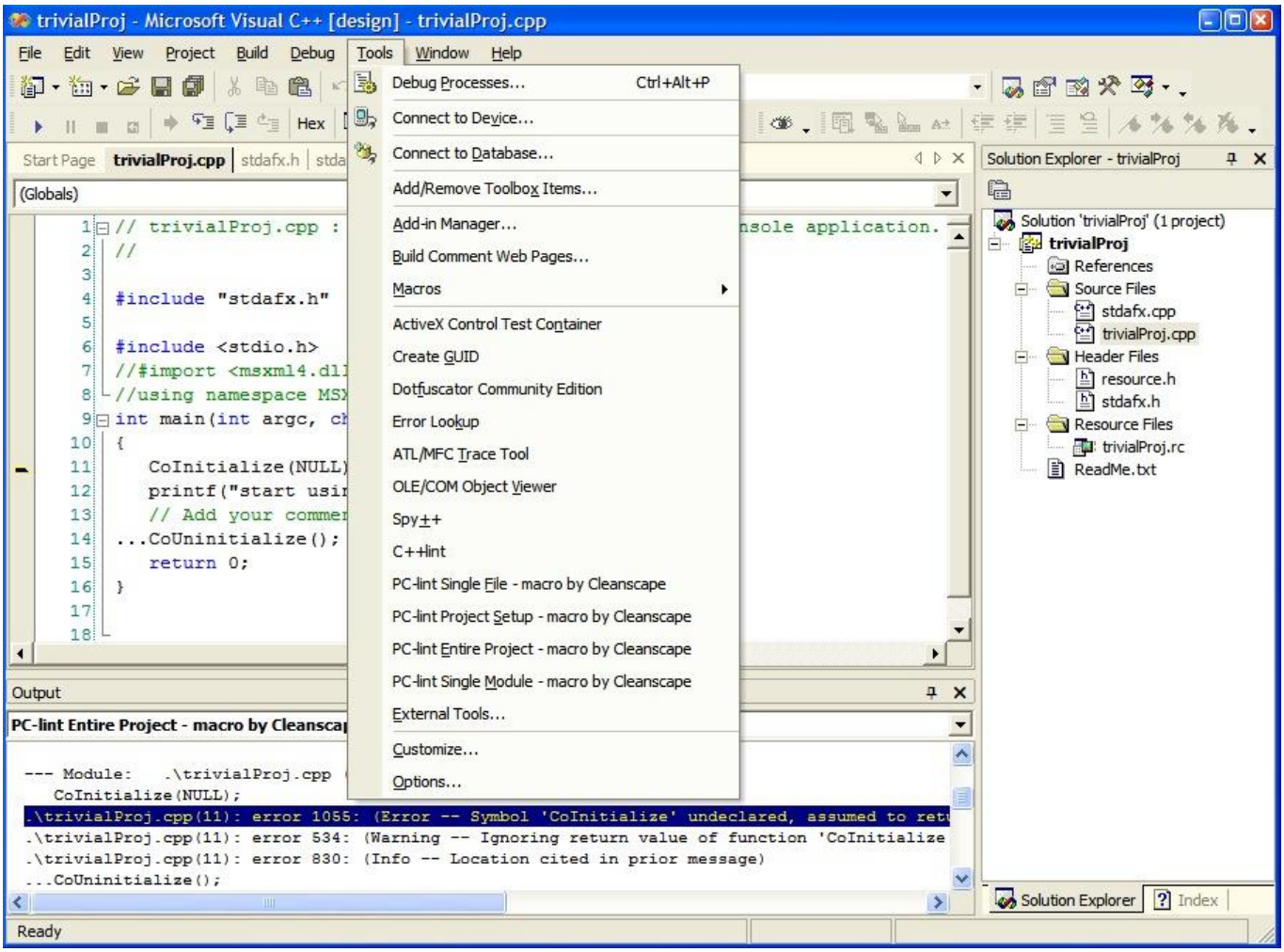
   Use this macro (after having set up the project by using tool #2 above) to run PC-lint over the *entire* project sourcebase. As with the single file tool (#1 above), analysis results appear in the Output window. Double-clicking on any error message will cause Visual Studio's built-in editor to jump to the line number in the source file causing the analysis message.

5. PC-lint Single Module - macro by Cleanscape

   Use this macro (after having set up the project by using tool #2 above) to run PC-lint over a single module. The difference between this tool and tool #1 is that all your project information is made available to the analysis. As with the single file and entire project tools (#1 and #4 above), analysis results appear in the Output window. Double-clicking on any error message will cause Visual Studio's built-in editor to jump to the line number in the source file causing the analysis message.

The screenshot on the next page shows the Cleanscape macros installed in Visual Studio's Tools dropdown menu. It also shows the analysis results of the current project. The highlighted (blue background) line in the Output window (bottom) was double clicked, resulting in Visual Studio's editor (top window) jumping to the source file / source line (line 11) that caused the analysis message.

The hyperlink feature also makes it very easy to add PC-lint "magic comments" to modify analysis at the statement or block level. For more information, see Sections 5.1 and 5.2 of the PC-Lint Reference Manual (last bookmark in the Acrobat .pdf version of this document).



D. Uninstallation

To uninstall these macos from Visual Studio, you can use the Uninstaller located in the "main" subdirectory. From a command prompt, type

```
uncpp –VS
```

Or, to manually remove any of them:

0. Open the Visual Studio IDE.
1. From the Tools dropdown menu, select "External Tools..." (in VS 6 select "Customize..." then click on the "Tools" tab in the resulting dialog).
2. Click the tool you wish to delete.
3. Click the "Delete" button in the right side of the dialog box (in VS6 click on the red 'X' in the upper right corner).

PART VI Running Cleanscape C++lint from the Command Line

A. Introduction

Cleanscape C++lint has a command line facility suitable for standalone operation or for inclusion in scripts, e.g., for "make lint" purposes.

For details on the actual operation of PC-lint and its control and reporting options, refer to the companion document, PC-Lint Reference Manual (last bookmark in the Acrobat .pdf version of this file).

B. Operation

To run C++lint in command line mode, you need to have set the environment variables as defined in Section 2.1.B.1.e or 2.2.B.f and registered the product as described in Section 3.

The format of the Cleanscape C++lint command line is quite simple:

```
cpplint <parameters_to_be_supplied_to_PC_lint>
```

Entering `cpplint` without parameters yields a command summary.

Details on all the command line parameters may be found starting in Chapter 4 of the PC-Lint Reference Manual (last bookmark in the Acrobat .pdf version of this file).

NEW Starting with version 1.3, you can use the Cleanscape GUI to generate analysis switches for you. To do so, set the analysis options within the GUI as described in Sections 4.B.5-7 and the Online Help. When done, select File – Save std.lnt as shown at right, and navigate to the directory in which you wish it to be saved. Some usage notes:

1. For compatibility with C++lint's Visual Studio mode, the GUI's "Target System" setting is stored within a comment. Uncomment this value when running from the command line, as described in the comments within the file.
2. If `std.lnt` already exists, you will be prompted before overwriting.
3. Any filenames entered in the GUI's project are ignored; the files that make a PC-lint project may be entered on the command line or appended to the `std.lnt` file one per line, with double-quotes surrounding filenames with spaces, as described in the comments within the file.

A sample `std.lnt` file is shown on the next page; yours may of course vary considerably.

```
//  *** This file created by Cleanscape C++lint GUI on 27-Jul-07 ***
// Refer to the PC-Lint Users Manual for details on control settings.

+fdi
+macros
+linebuf
+linebuf
-width(0)
-zero(400)
-passes(2)
-iC:\PROGRA~1\cleanscape\cpplint\main\lbin\lf
-iC:\PROGRA~1\cleanscape\cpplint\main\lbin\

// Uncomment next line for command line (non-IDE) operation:
// co-msc70.lnt

-iC:\PROGRA~1\cleanscape\cpplint\main\lbin\lf\ansi
-iC:\temp\C__~1
+fcp
+fan
+fbo
-u
-wlib(1)
-A
-w3
-d__WIN32
-e789

// For command line operation, list the C/C++ sourcefiles to be analyzed
// below, one per line.  Enclose in double quotes if name contains spaces.
```

C. Return Codes

A return code of zero (0) indicates that Cleanscape C++lint ran and ran successfully without encountering any source errors.

A return code >1 indicates that either

- There was a problem securing a valid license key to run the program, or
- There were one or more messages resulting from the PC-lint analysis over the source code.

A description of each result is of course available in the analysis report. For a complete list of analyses by number – including a link to detailed explanations, see http://www.cleanscape.net/products/cpplchecks.html

If there was a problem starting the program or securing a key, contact Cleanscape Technical Support at support@cleanscape.net. If you are under maintenance, you may also contact Cleanscape Support for questions regarding any analysis output message.

For more information PC-lint's return codes and their uses, see Section 4.2 of the PC-Lint Reference Manual (last bookmark in the Acrobat .pdf version of this file).

Of particular note in that section is the possible use of the –zero (#) feature, which can keep the return code at zero when warnings – but not errors – are encountered during the analysis. For this mode, –zero(400) is recommended.

PART VII MISCELLANEOUS INFORMATION

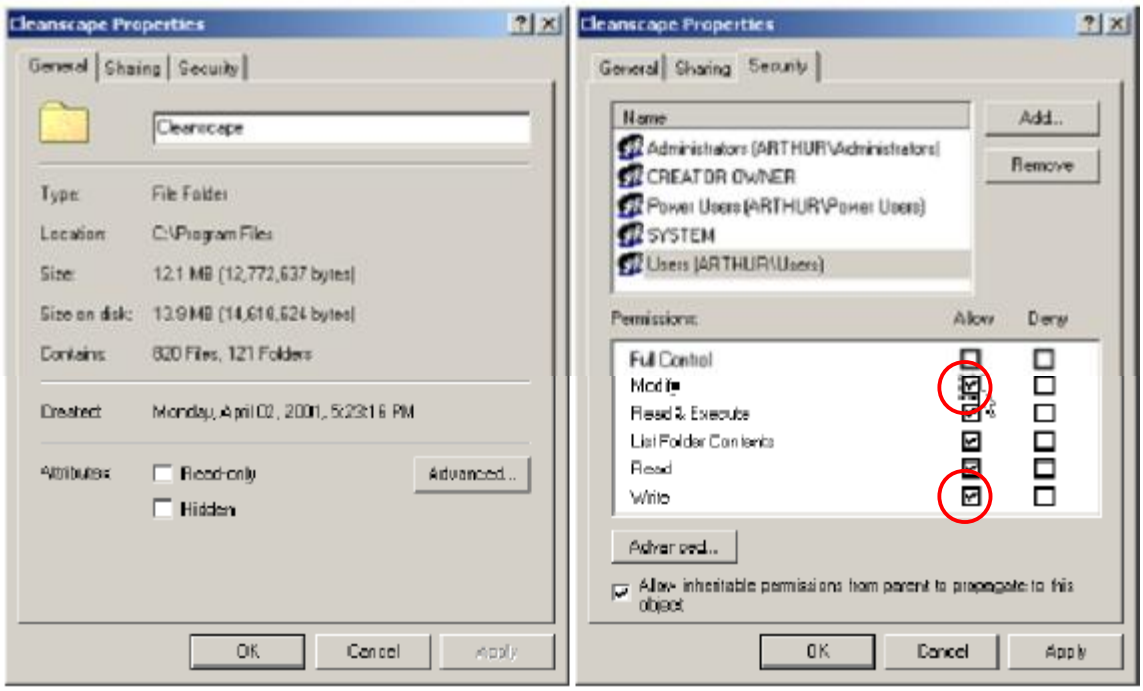
7.1 ADDITIONAL STEPS FOR WINDOWS 2000

A. Important note

1. This section applies to users running Windows 2000 who belong to the "Users" group, and only to that group.

B. Details

1. For Cleanscape C++lint to run correctly under Win2k, users must have "write" and "modify" access rights to the installation directory and all its subdirectories. This section explains the procedure used to change the access rights described above.
   a. Log in as "administrator" and finish installing Cleanscape C++lint.
   b. Double-click on the "My Computer" icon on the desktop.
   c. Navigate to and double-click on the installation folder. Select Properties from the sub-menu.
   d. Select "Security" tab on the Properties screen:



   e. Select the "Users" group and enable "Modify" and "Write" permissions.
   f. Click the "Apply" button.
   g. Click the "OK" button. This should close the Properties window.
   h. Cleanscape C++lint is now ready to run on Win2k for the "Users" group.

7.2 ADDING AN EXTERNAL EDITOR TO THE GUI USING SETEDITOR

A. Introduction

By popular demand, Cleanscape has added the ability for users to specify their own favorite editor to any Cleanscape product (as opposed to submitting a feature request to Cleanscape Support). This is implemented via an external program called seteditor, located in the 'bin' subdirectory.

User contributions welcome! Send them to support@cleanscape.net; any contributions will receive appropriate credit and be placed in a "master" file located at http://www.cleanscape.net/products/contributed_editors.html.

B. Operation

Windows.

You can either run seteditor from the command line or via Explorer.

From a DOS shell (cmd or command prompt), run the following command:
`"<install_dir>\bin\seteditor"`

From Explorer, navigate to the above directory and then double-click `seteditor.exe`.

Unix.

From a shell prompt, run the following command:
`<install_dir>/bin/seteditor`

Three pop-up dialogs (Windows) or a sequence of shell interactions (Unix/Linux) will guide you through

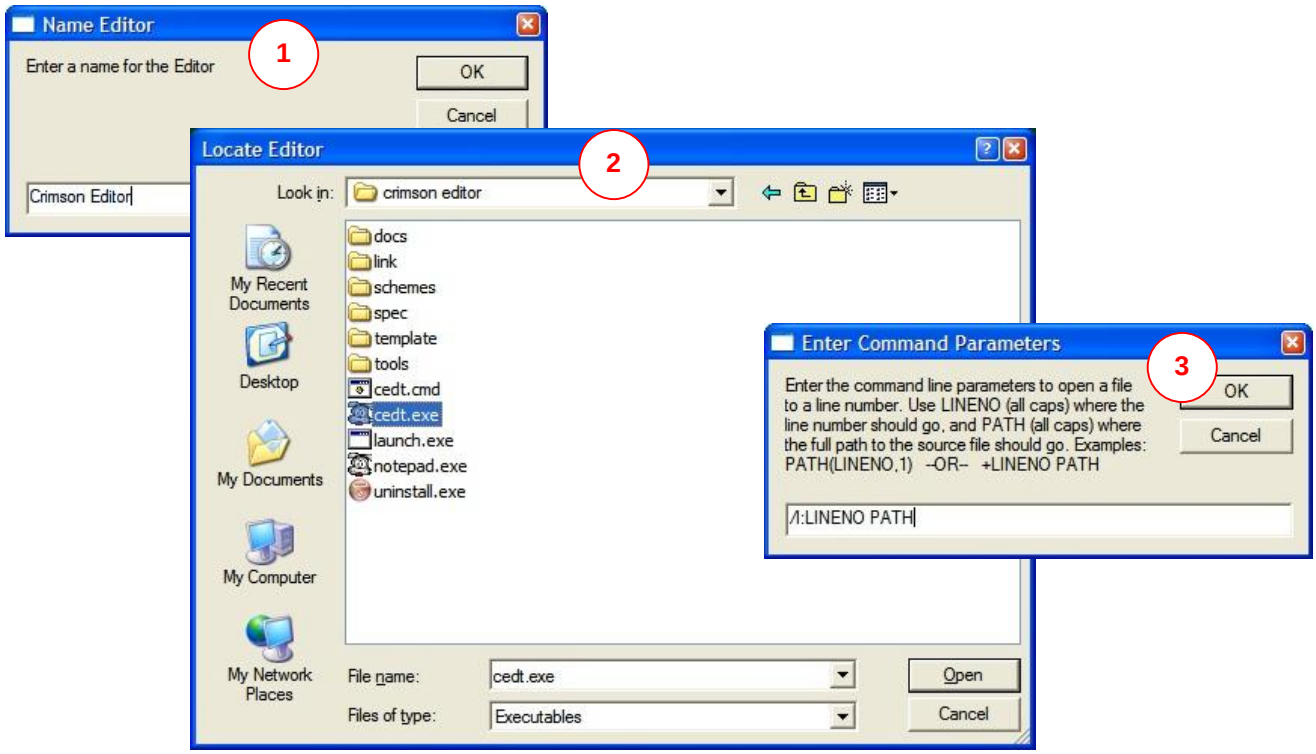
1. Naming the editor (a label identifier)
2. Locating the editor executable itself
3. Setting command line parameters to open a file and jump to a line number.

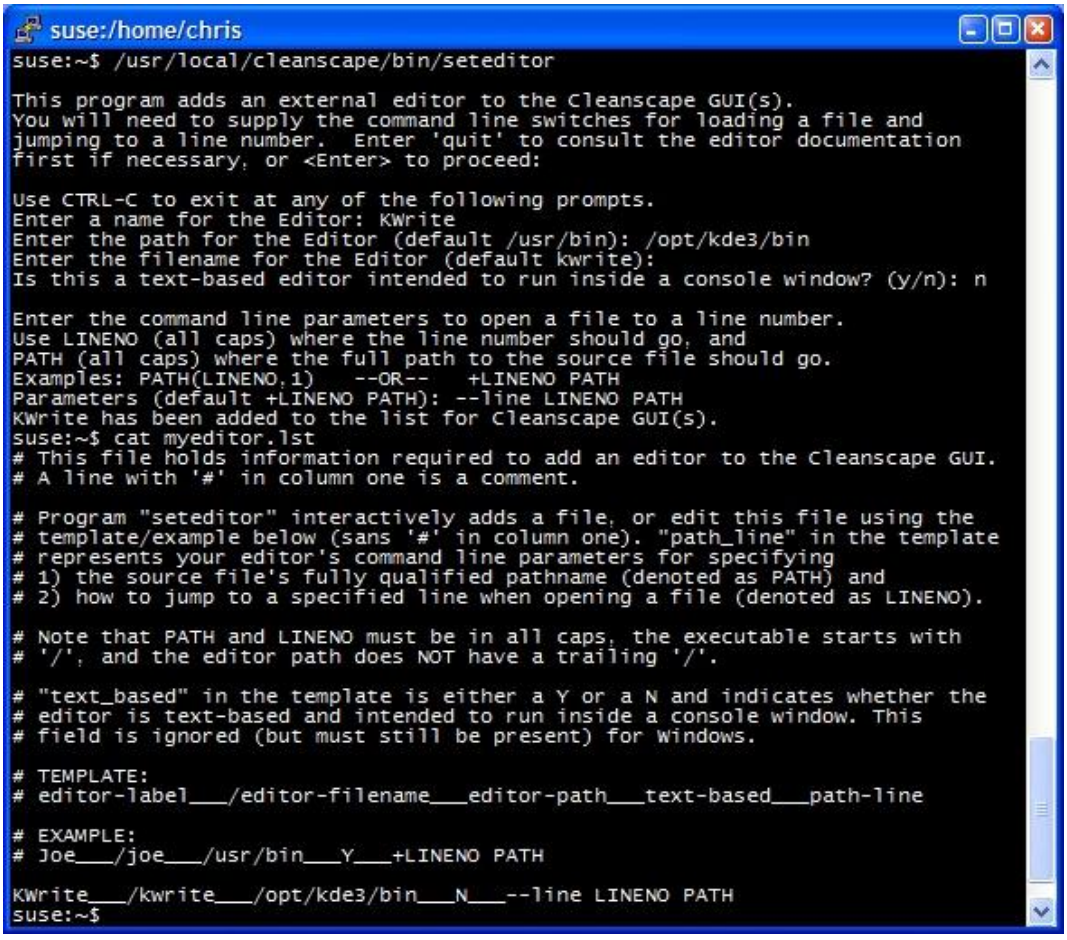
A sample Windows session depicting the dialogs for all three steps (and labeled as such) is shown on the next page, as is a Unix/Linux shell session.

NOTE: Refer to your editor's documentation to get the editor's command line information required (i.e., specifying the filename to open and the line number to jump to when opening the file). If your editor does not support jumping to line numbers from the command line, you can still invoke the editor but it will be impossible to align the analysis message to the "offending" source line.

Any number of editors may be added in this fashion. Added file information is stored in file `myeditor.lst`, located in bin subdirectory on Windows or your `$HOME` directory if running Unix/Linux. Once successfully added, email your `myeditor.lst` file to support@cleanscape.net for inclusion in a Master file to share with other Cleanscape customers!

It is also possible to edit `myeditor.lst` manually; see the comments inside the file. The Unix/Linux session also shows the contents of `myeditor.lst` (which looks substantially similar under Windows).


Name Editor
1
Enter a name for the Editor
OK
Cancel
Crimson Editor
Locate Editor
2
Look in:
crimson editor
My Recent Documents
Desktop
My Documents
My Computer
My Network Places
docs
link
schemes
spec
template
tools
cedt.cmd
cedt.exe
launch.exe
notepad.exe
uninstall.exe
File name:
cedt.exe
Open
Files of type:
Executables
Cancel
Enter Command Parameters
3
Enter the command line parameters to open a file to a line number. Use LINENO (all caps) where the line number should go, and PATH (all caps) where the full path to the source file should go. Examples: PATH(LINENO,1) -OR- +LINENO PATH
OK
Cancel
/I:LINENO PATH


suse:/home/chris
suse:~$ /usr/local/cleanscape/bin/seteditor
This program adds an external editor to the Cleanscape GUI(s).
You will need to supply the command line switches for loading a file and
jumping to a line number. Enter 'quit' to consult the editor documentation
first if necessary, or <Enter> to proceed:
Use CTRL-C to exit at any of the following prompts.
Enter a name for the Editor: KWrite
Enter the path for the Editor (default /usr/bin): /opt/kde3/bin
Enter the filename for the Editor (default kwrite):
Is this a text-based editor intended to run inside a console window? (y/n): n
Enter the command line parameters to open a file to a line number.
Use LINENO (all caps) where the line number should go, and
PATH (all caps) where the full path to the source file should go.
Examples: PATH(LINENO,1) --OR-- +LINENO PATH
Parameters (default +LINENO PATH): --line LINENO PATH
KWrite has been added to the list for Cleanscape GUI(s).
suse:~$ cat myeditor.lst
This file holds information required to add an editor to the Cleanscape GUI.
A line with '#' in column one is a comment.
Program "seteditor" interactively adds a file, or edit this file using the
template/example below (sans '#' in column one). "path_line" in the template
represents your editor's command line parameters for specifying
1) the source file's fully qualified pathname (denoted as PATH) and
2) how to jump to a specified line when opening a file (denoted as LINENO).
Note that PATH and LINENO must be in all caps, the executable starts with
'/', and the editor path does NOT have a trailing '/'.
"text_based" in the template is either a Y or a N and indicates whether the
editor is text-based and intended to run inside a console window. This
field is ignored (but must still be present) for Windows.
TEMPLATE:
editor-label___/editor-filename___editor-path___text-based___path-line
EXAMPLE:
Joe___/joe___/usr/bin___Y___+LINENO PATH
KWrite___/kwrite___/opt/kde3/bin___N___--line LINENO PATH
suse:~$