

CodeMeasure User's Guide



Version 1.3

a product of

S.A.F.E.

**Software Analysis & Forensic Engineering
Corporation**

Table of Contents

| | |
|---------------------------------------|----|
| Introduction | 1 |
| About CodeMeasure® | 1 |
| Copyrights, Trademarks, Patents | 1 |
| Copyrights | 1 |
| Trademarks | 1 |
| Using CodeMeasure..... | 2 |
| System Requirements | 2 |
| The User Interface | 2 |
| Menu | 2 |
| Source Code Folders | 4 |
| Compare Radio Buttons..... | 4 |
| File types..... | 4 |
| Buttons..... | 5 |
| Status Bar | 5 |
| Running CodeMeasure | 5 |
| Step 1..... | 6 |
| Step 2..... | 6 |
| Step 3..... | 6 |
| Step 4..... | 6 |
| CodeMeasure Spreadsheet..... | 7 |
| Advanced Topics..... | 10 |
| Command line interface | 10 |
| #..... | 10 |
| [Versions] | 10 |
| [FileTypes] | 10 |
| [CompareTo] | 10 |
| Support..... | 12 |

Introduction

About CodeMeasure®

CodeMeasure is a software development utility used to measure development progress across two or more versions of software. CodeMeasure provides a quantitative analysis of your source code and calculates growth metrics that give you a clear picture of the amount of development as a percentage of the original version or as a percentage of the previous version of the code.

CodeMeasure provides developers and managers with an easy approach to measure code evolution over time and correlate the amount of code changed to the effort required for development. CodeMeasure delivers consistent and accurate results regardless of programming language in a simple and easy-to-use package. CodeMeasure will give you the data you need to add credibility to your development schedule plans and give you an easy way to measure development progress once your project is underway.

Copyrights, Trademarks, Patents

Copyrights

The materials in this users guide are copyright 2009-2010 by Software Analysis and Forensic Engineering Corporation (SAFE Corporation).

All written materials from SAFE Corporation regarding CodeMeasure, including the material in this User's Guide and the source code for all versions of CodeMeasure are the intellectual property of SAFE Corporation.

Trademarks

SAFE Corporation, the SAFE Corporation logo, the SAFE Corporation brand, CodeMeasure, the CodeMeasure logo, and all other SAFE Corporation product names referenced herein are registered trademarks or trademarks of SAFE Corporation. All other brand and product names mentioned herein are trademarks of their respective owners.

Using CodeMeasure

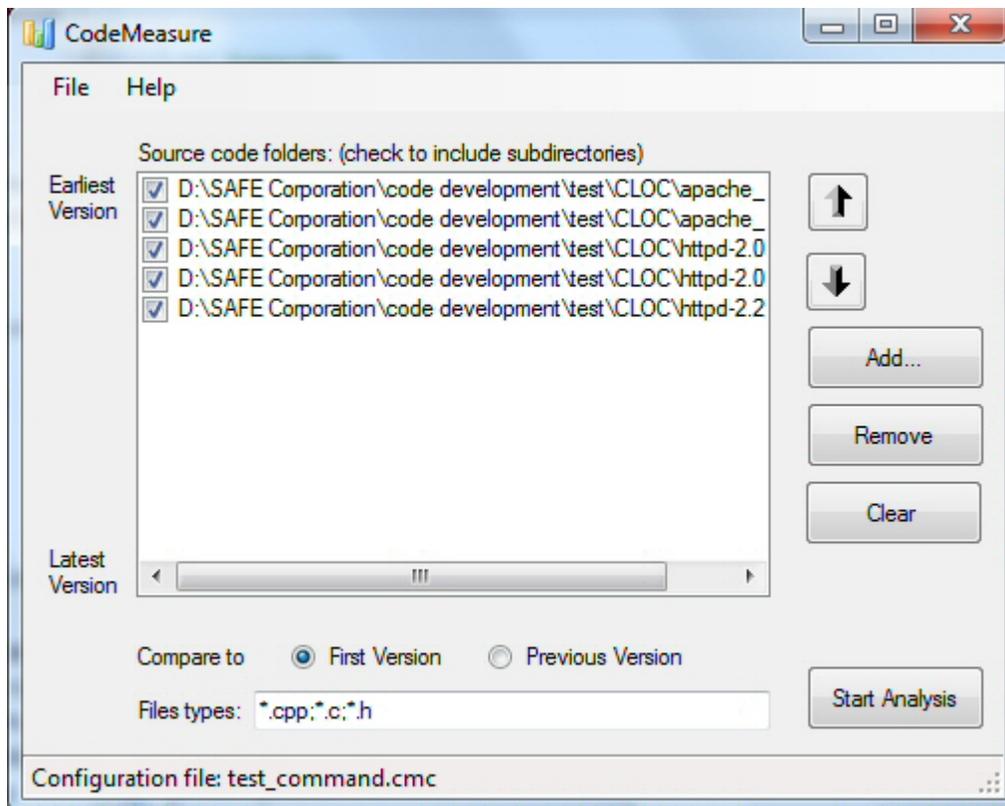
System Requirements

CodeMeasure will run on any computer using any of the following versions of the Microsoft Windows operating system:

- Windows 2000
- Windows XP
- Windows Vista
- Windows 7
- Windows 10
- Windows 11

The User Interface

CodeMeasure presents a simple graphical user interface shown below.



Menu

The CodeMeasure menu commands are described below.

File Menu

The File menu options are described below.

Open

Open a CodeMeasure configuration file. This allows you to restore settings for a CodeMeasure run.

Save

Save a CodeMeasure configuration file. This allows you to save settings for a CodeMeasure run.

Save As

Save a CodeMeasure configuration file with a new name. This allows you to save settings for a CodeMeasure run.

Exit

Exit the CodeMeasure program.

Help Menu

The Help menu options are described below.

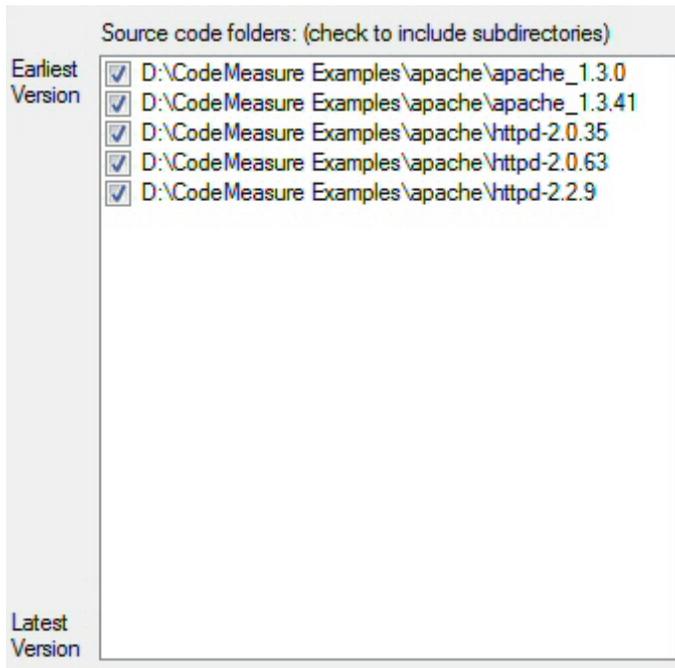
About

Display information about the CodeMeasure program.

Help

Open this help file.

Source Code Folders



The source code folders region lists all of the source directories to compare. The check box is selected by default and tells CodeMeasure to examine files in all the subdirectories of that source folder. Uncheck the box if only the top level directory should be examined. The order of the source folders is significant; the first folder in the list is the earliest version of the code and the last first folder in the list is the latest version of the source.

Compare Radio Buttons



The compare radio buttons define whether you want to compare all source code to the first version of the source code or compare the source code to the previous version of the source code. The first version is found in the first folder listed in the source folders window and the previous version is defined as the source folder immediately preceding the folder being analyzed.

File types



The file types text box allows entry of the files or file types that will be compared. In this example CodeMeasure will compare all files with ".cpp; .c; or .h" file extensions. Each different type needs to be separated by a semicolon.

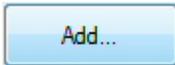
Buttons



The up arrow button will move the selected folder up one position in the list.



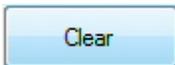
The down arrow button will move the selected folder down one position in the list.



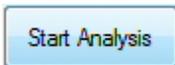
The "Add" button will open a file browser and allow you to select a source code folder to analyze. Once selected, the folder will be added to the end of the list. To reposition the folder select the folder and use the arrow keys.



The "Remove" button will remove the selected folder from the list.

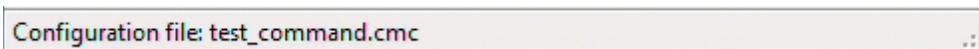


The "Clear" button will remove all folder entries from the list.



The "Start Analysis" button signals CodeMeasure to begin analysis of the files in the directories and optionally all subdirectories if specified.

Status Bar



The status bar shows the current open CodeMeasure configuration file, if any.

Running CodeMeasure

To run CodeMeasure is a simple four step process. See the [User Interface](#) section for more information.

Step 1

Select the folders containing the versions of software to be compared. Use the arrows to order them from earliest (top) to latest (bottom). Uncheck the boxes if you don't want to include subdirectories.

Step 2

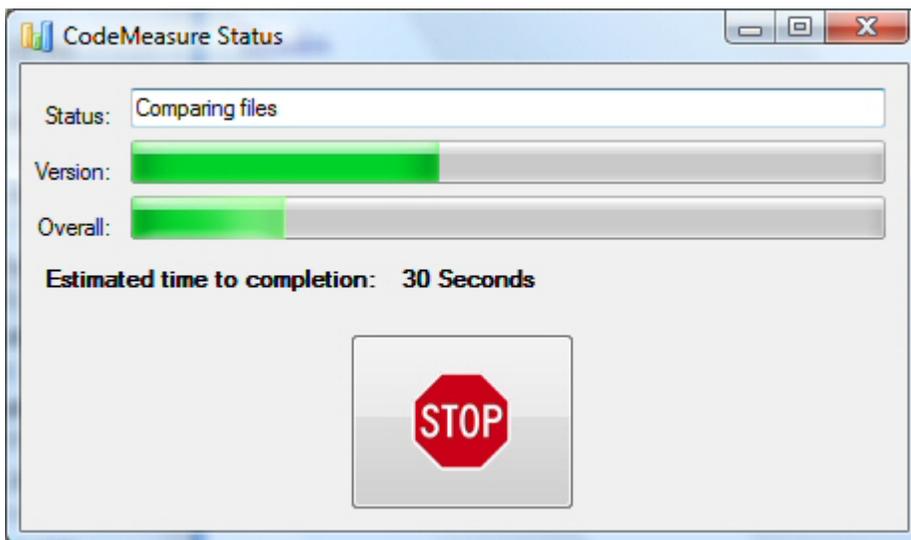
Select whether you wish to compare each software version to the first version or to the previous version.

Step 3

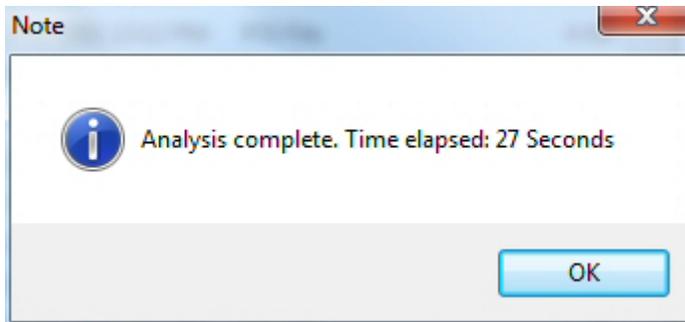
Enter the file types separated by semicolons (e.g., *.cpp;*.c;*.h).

Step 4

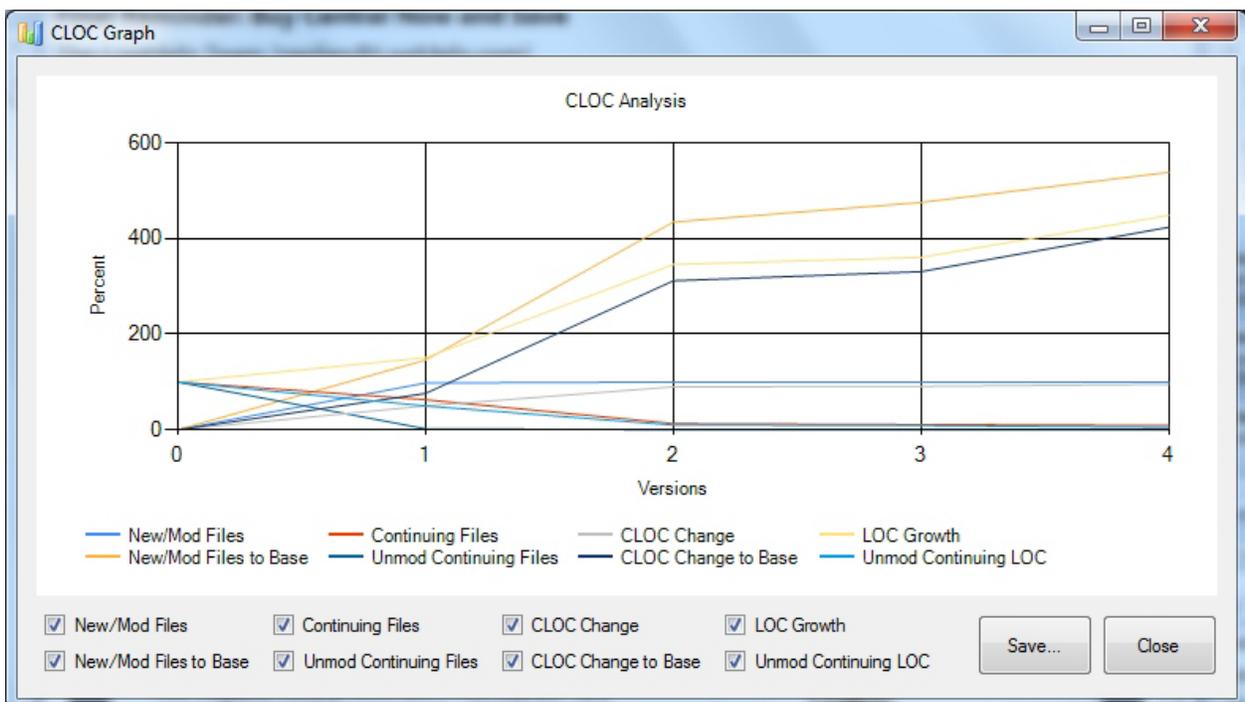
Press the Start Analysis button to start the analysis. You will see a status window pop-up, shown below, to give you the progress of the analysis.



Press the Stop button to abort the analysis. When CodeMeasure completes, a notification window, shown below, will pop-up.



After clicking "OK," a graph of various CLOC percentages will appear as shown below.



Clicking on the checkboxes at the bottom turns on and off the various lines on the graph. Clicking "SAVE" brings up a window asking you to name the output file for your analysis. The output spreadsheet file will then be opened in your default spreadsheet program.

CodeMeasure Spreadsheet

CodeMeasure generates a comma separated value (.csv) spreadsheet file. A sample file is shown below.

CLOC Spreadsheet

Date: 2/14/2010
Time: 8:18:09
Version 0 directory: D:\SAFE Corporation\code development\test\CLOC\mozilla0.1 and subdirectories
Version 1 directory: D:\SAFE Corporation\code development\test\CLOC\mozilla0.8 and subdirectories
Version 2 directory: D:\SAFE Corporation\code development\test\CLOC\mozilla1.0 and subdirectories
Version 3 directory: D:\SAFE Corporation\code development\test\CLOC\mozilla1.5 and subdirectories
Version 4 directory: D:\SAFE Corporation\code development\test\CLOC\mozilla2.0 and subdirectories
Version 5 directory: D:\SAFE Corporation\code development\test\CLOC\mozilla3.0 and subdirectories
File types: *.c;*.cpp;*.h
Compare: Each version to the initial version

| | Version 1 | Version 2 | Version 3 |
|--|-----------|-----------|-----------|
| Version Summary | | | |
| Total files [TF] | 2023 | 2307 | 2703 |
| Total lines of code [LOC] | 486209 | 539705 | 653037 |
| Total bytes [TB] | 22124474 | 24464843 | 30209333 |
| Examined Version [EV] | Version 1 | Version 2 | Version 3 |
| Base Version [BV] | Version 1 | Version 1 | Version 1 |
| Change Summary | | | |
| Files | | | |
| New files [NF] | 0 | 303 | 701 |
| Modified continuing files [MCF] | 0 | 1957 | 1957 |
| Lines | | | |
| New and modified lines of code [CLOC] | 0 | 106845 | 224042 |
| Continuation Summary | | | |
| Files | | | |
| Continuing files [CF] | 2023 | 2004 | 2002 |
| Unchanged continuing files [UCF] | 2023 | 47 | 45 |
| Lines | | | |
| LOC in continuing files [LOCinCF] | 486209 | 480595 | 487041 |
| CLOC in continuing files [CLOCinCF] | 0 | 47735 | 58046 |
| Unchanged continuing lines of code [ULOC] | 486209 | 432860 | 428995 |
| Summary Statistics | | | |
| Files | | | |
| Percent of new and modified files ((NF+MCF)/TF) | 0% | 98% | 98% |
| Percent of new and modified files relative to base version ((NF+MCF)/TF[BV]) | 0% | 112% | 131% |
| Percent of continuing files (CF/TF) | 100% | 87% | 74% |
| Percent of unchanged continuing files (UCF/TF) | 100% | 2% | 2% |
| Lines | | | |
| Percent CLOC change (CLOC/LOC) | 0% | 20% | 34% |
| Percent CLOC change relative to base version (CLOC/LOC[BV]) | 0% | 22% | 46% |
| Percent LOC growth (LOC/LOC[BV]) | 100% | 111% | 121% |
| Percent unchanged continuing lines of code (ULOC/LOC) | 100% | 80% | 66% |

The header of the spreadsheet provides all the information about the analysis: date, time, directories for each version analyzed and the file types compared. The report section contains the results of the analysis and is explained below.

Total files [TF]: The total number of files in the version being examined.

Total lines of code [LOC]: The total lines of code (blank lines ignored) in the version being examined.

Total bytes [TB]: The total bytes of code in the version being examined.

Examined version [EV]: The version being examined.

Base version [BV]: The version against with the version being examined is compared. This is either the first version or the previous version, depending on which was selected.

New files [NF]: The number of files that exist in the examined version but did not exist in the base version.

Modified continuing files [MCF]: The number of continuing files (files that exist in both the examined version and the base version) with new lines of code.

New and modified lines of code [CLOC]: The number of lines of code in the examined version that are new or have been modified from the base version.

Continuing files [CF]: The number of files in the examined version that were also in the base version.

Unchanged continuing files [UCF]: The number of continuing files that have not changed from the base version to the examined version.

Lines of code in continuing files [LOCinCF]: The number of lines of code in the continuing files.

New and modified lines of code in continuing files [CLOCinCF]: The number of lines of code in the continuing files that are new or have been modified from the base version.

Unchanged continuing lines of code [ULOC]: The number of lines of code in continuing files that have not changed.

Percent of new and modified files: $((NF+MCF)/TF)$

Percent of new and modified files relative to base version: $((NF+MCF)/TF[BV])$

Percent of continuing files: (CF/TF)

Percent of unchanged continuing files: (UCF/TF)

Percent CLOC change: $(CLOC/LOC)$

Percent CLOC change relative to base version: $(CLOC/LOC[BV])$

Percent LOC growth: $(LOC/LOC[BV])$

Percent unchanged continuing lines of code: $(ULOC/LOC)$

Advanced Topics

Command line interface

CodeMeasure can also be run from the command line or called from a batch file or script. To run CodeMeasure type the following at the command shell prompt:

```
>"C:\Program Files\SAFE\CodeMeasure\CodeMeasure.exe" configure.cmc output.csv
```

The parameter `configure.cmc` is the name of a CodeMeasure configuration file. The parameter `output.csv` is the name of the CSV format spreadsheet for the results of the analysis. The format of the configuration file is as follows:

#

Any line starting with `#` denotes a comment line

[Versions]

This tag identifies that the lines following, list the directories to be analyzed from earliest to latest.

```
[Versions]
1 "c:\TEMP\Version0"
0 "C:\TEMP\Version1"
1 "C:\TEMP\Version2"
```

Where the 1 or 0 enables or disables the inclusion of subdirectories, followed by the path to a version of the code to be examined.

[FileTypes]

This tag identifies that the next line lists the file types to be examined. Each file type identified is separated from the previous with a semicolon ";".

```
[FileTypes]
*.cpp;*.h;*.c
```

[CompareTo]

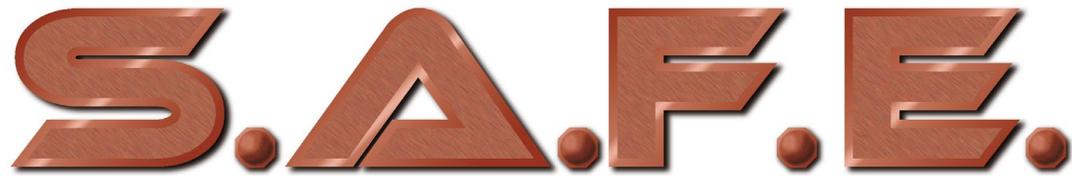
This tag identifies that the next line defines whether you want to compare each version of code to the "first" version in the [Versions] list or to the "previous" version.

```
[CompareTo]
previous
```

A sample configuration file is shown below:

```
# CodeMeasure command file
[Versions]
1 .\addtest\Orig
1 .\addtest\New
[FileTypes]
*. *
[CompareTo]
first
```

Support



Software Analysis and Forensic Engineering Corporation
Web: www.SAFE-corp.com
Email: Support@SAFE-corp.com