



Source Crusade

Leading the fight for better code



<p>Project: Source Metrix Title : User Manual</p>

Compiled by : Bruce S. O. Adams

Date : 10-Apr-2006

Issue	Date	Author	Change
1.1	10-Apr-2006	BSA	Updated for initial preview release
1.0	12-Jan-2006	BSA	Initial Draft



1 INTRODUCTION3

1.1 WHAT IS SOURCE METRIX?3

1.2 REGISTRATION3

1.3 VERSION.....3

1.4 REQUIREMENTS.....3

1.5 CONTACT DETAILS.....3

2 INSTALLATION4

3 QUICK START GUIDE4

4 HOW IT WORKS4

5 SOURCE FILES.....4

5.1 SOURCE TYPES.....4

5.2 DEFAULT FILE EXTENSIONS4

6 DATABASE5

6.1 ANALYSING CHANGES WITHOUT STORING THEM.....5

7 METRICS5

7.1 LINE BASED METRICS5

7.2 STATEMENT BASED METRICS5

7.3 DENSITY5

7.4 FUTURE ENHANCEMENTS6

8 REPORTING FUNCTIONALITY6

8.1 FUTURE ENHANCEMENTS6

9 INTERFACING WITH VERSION CONTROL6

9.1 SUPPORTED VERSION CONTROL SYSTEMS.....6

10 HISTORICAL ANALYSIS6

10.1 WHY USE HISTORICAL ANALYSIS FEATURES?6

10.2 HOW TO USE THE HISTORICAL ANALYSIS FEATURE.....6

10.3 AVERAGE CHANGE.....7

10.4 PREDICTION.....7

10.4.1 *Future developments*.....7

11 CHANGE GROUPING.....7

11.1 WHAT IS A CHANGE GROUP?.....7

11.2 HOW ARE CHANGE GROUPS DETERMINED?7

11.3 HOW DO I USE THIS FEATURE?.....8

11.4 STORING AND OVERRIDING CHANGE GROUPS8

12 COMMAND LINE OPTIONS SUMMARY8

12.1 CHANGE GROUPING OPTIONS.....9

13 PERFORMANCE.....10

13.1 FIRST USE10

13.2 OVERSIGHT THREAD10

13.3 NETWORK DATABASES10

13.4 FUTURE ENHANCEMENTS10

14 APPENDIX A: CHANGE GROUPING TEXT FORMAT.....11

15 LEGAL BLURB12

Introduction

Thank you for using Source Metrix.
This manual provides detailed instructions on how to use the software.

Source Metrix development is ongoing. If you have any suggestions on how it may be improved or in the event you find any problems please report them via the contact details given below.

What is Source Metrix?

Source Metrix is a tool to analyse your source code and calculate metrics that describe various properties of it.

The information gathered is designed to be of use to both engineers and project managers. Source Metrix is able to measure the size of a software project and the size of each change made to it.

By analysing the history of a project from your version control archives Source Metrix is also capable of providing estimates of how long new changes may take to develop.

Source Metrix is also capable of providing metrics that give an indication of the quality of each source file, thus helping to identify areas that need more work.

Registration

Source Metrix represents many hours of development time to provide you with the highest quality product. Please help this effort continue by registering your software online at www.sourcecrusade.com. In order to encourage registration the product is licensed for an evaluation period of 30 days from first use after which it will cease to function. If you are still unsure after this time we can arrange an extension to this period.

At Source Crusade we believe software to be more a kind of service industry rather than a manufacturing one. This means that when you register you are paying not just for what you already have but for any improvements we develop during the lease period. This represents much better value for money than when you buy a shrink-wrapped product off the shelf.

As a registered user your voice will also have greater influence over what enhancements are developed and what other products we bring into production.

Version

This documentation is up to date for Source Metrix version 0.999.6 issued on 10th-April-2006

Requirements

Source Metrix should run on any windows system with no special requirements.

A linux version is currently under development.

Currently only English language is supported.

If you are interested in obtaining a copy for another platform or language (either human or computer) please contact us with details.

Contact Details

Home page: www.sourcecrusade.com
suggestions@sourcecrusade.com
bugs@sourcecrusade.com
customerservices@sourcecrusade.com



Installation

Simply copy the Source Metrix executable (srcmetrix.exe) and data file (srcmetrix.dat) to a suitable directory and ensure that your path variable is updated to point at it.

E.g. from a DOS prompt.

```
mkdir c:\tools
copy srcmetrix.exe c:\tools
copy srcmetrix.dat c:\tools
set PATH=%PATH%;c:\tools
```

Quick Start Guide

Source Metrix has been designed to be as simple to use as possible. For the most basic operations all you have to do is start a DOS window and go to the directory containing the source you wish to process and type "srcmetrix". The tool will automatically search for and analyse any source files it finds.

The information gathered will be reported immediately and also stored in a database file (by default "srcmetrix.db").

On subsequent runs the software will compare the files against the stored information and provide additional information on the changes made.

How it works

Source Metrix analyses your source files and optionally stores relevant statistics in a database.

The information thus gathered is used to generate a report.

Certain features, such as historical analysis, require read only access to your version control system if available.

There are several elements that can be controlled by the user:

- What source files will be analysed
- What information is required in the reports
- What database to use and how it should be updated
- How to talk to the version control system

Source Files

The tool considers sources files to be organised into projects.

The project name is set using the -project option

Once a file has been added to a project the tool knows to include it in all summary reports thereafter and to analyse it when not explicitly stated.

By default the tool uses the list of files associated with the project name in the database.

If no associations have been made then the tool searches the current directory for all files with appropriate extensions and analyses them.

Source Types

Source Metrix has been designed to work with C and C++.

It may work unmodified with other C-like languages but is untested. Your mileage may vary.

Formal support for other languages including Java, D & C# is being considered for a future version.

Default File Extensions

When searching for source files the tool assumes the following to be relationships:

File Extension	Source Type
----------------	-------------



.cpp	C++ source
.cxx	C++ source
.c	C source
.h	C or C++ header
.hpp	C++ header

However, any file name specified on the command will be treated as source.

Database

A database can be used to store the results of file analysis.

The database is also used to cache data gathered previously. This speeds up reporting for files analysed previously that have not changed.

By default the tool will look for and if necessary create a database called SrcMetrix.db

If the `-project` option is used the default database name will be `<projectName>.db`

The name of the database may be specified on the command line using the `-database` option

Analysing Changes without storing them

A typical use for the tool is to analyse a proposed change to the code base before it is integrated.

In this circumstance you want the tool to report the changes made but not to store the results in the database.

The `-nocommit` option will prevent the tool from storing results in the database.

Metrics

This section describes the metrics that the software currently supports.

Line based Metrics

Files are parsed line by line and each line is categorised as either code, comment or blank.

- Source lines of code
- Comments lines
- Blank lines
- 'Physical' Comment density – ratio of comment lines to code lines
- 'Physical' Code density – ratio of code lines to blank & comment lines

Statement based Metrics

Files are parsed and split into statements. Each statement is categorised as either code, comment or blank.

- Source statements
- Comment statements
- Blank statements
- 'Logical' comment density – ratio of comment lines to code lines
- 'Logical' code density – ratio of code statements to blank & comment statements

Density

Density list metrics above indicate the ratio of code to comments. These should fall in reasonable bounds. A low ratio indicates that the code is probably insufficiently commented. A very high ratio suggests that there is code missing or that commenting has been over zealous.

Future Enhancements

In the near future the tool will be updated to allow lower and upper bounds for metrics to be set. If these are exceeded quality warnings can be issued.

In the future we intend to add many more metrics including some giving an idea of quality in terms of code complexity and readability.

Reporting Functionality

By default reports are generated to the command line.

Reports may be redirected to a file using the `--report` option.

The content of reports is affected by the following options:

- changes - reports will be generated only for files that have been modified since the last run
- summary – display summary only not details for individual files
- reportall – report all metrics including those previous calculated for unmodified files

Future Enhancements

Export to excel, csv & XML is tabled release in the near future.

Integration with ERP (Enterprise Resource Planning) systems is tabled for a subsequent release.

Interfacing with Version Control

You may specify a particular version control system using the `--vcs` option.

If you do not specify a particular VCS but specify an option that requires its use the tool will attempt to guess the system in use based on environment variables, commands available and the local directory structure.

Supported Version Control systems

We currently support the following version control systems:

- CVS
- PVCS

Support for additional systems is planned in the future.

If your version control system is not specified above please contact us indicating your needs.

Historical Analysis

Why use historical analysis features?

By interrogating your version control archives the tool is able to obtain previous versions of your files and analyse them.

This helps to show how your code base has changed over time.

This information can in turn be used to help in making predictions.

It can also be used to help determine whether a new change is consistent with the previous changes made to the codebase.

For example if a typical code change for your project is 100 lines across 5 files you might be suspicious of an unusually large change such as 1000 lines across 10 files. If it was recent was it reviewed thoroughly enough and could it be a potential source of bugs, including one you are currently trying to pin down. You might be even more suspicious if this unusually large change also contains much fewer comments than normal.

How to use the historical analysis feature

Specify the `--history` command line option to analyse the history of an individual file.

Use the `--averagechange` option to determine the size of an average change for your project.

Use the –predict option to estimate how long it might take to do X new changes based on your previous performance.

Average Change

The tool is able to generate a description of an “average change”.

This includes metrics such as how many lines it consists of and how many files it would affect.

The average change is determined by combining historical analyses with the change grouping feature described below.

In practice changes will range from much bigger than average right down to single line changes. However on a large project the average change can be a useful device.

In a future version we intend to improve on this by allowing for a number of characteristic ‘average’ changes which can be representative of distinct activities within your project development (e.g. new features versus bug fixes)

Prediction

For planning purposes the time between changes can be used to estimate how long each change took.

The tool can make predications based on the average change, how long it took and the number of changes required.

This of course cannot take into account information not included such as when people were in meetings or on holiday but can serve as a useful guide. Accuracy will increase with the length of time considered and the size of the project.

1.1.1 Future developments

In future versions we intend to improve the tools predictive abilities as follows:

- indications of statistical accuracy.
- Monte carlo analysis over a range of change types
- Taking other data into account such as the rate at which new bugs and feature requests occur.

Change Grouping

What is a change group?

Version control systems are not generally geared to the purpose of analysing old source code. Typically they just store it for future reference in case you need to back track or determine how a particular change came to be.

Most version control systems are organised around the notion of file deltas. A file delta is a change to an individual file.

By contrast development tends to consist of groups of changes to several files more or less simultaneously.

The change grouping feature of the tool allows the organisation of file deltas into groups to be recovered from version control systems that do not explicitly record it.

How are change groups determined?

The tool has several ways of determining whether two deltas belong to the same change group.

Changes made at a similar time by the same author are likely to be for the same purpose.

Changes made which have the same change comment are likely to be for the same purpose.

E.g.

FileA	revision 1.2	by Alice at 12:30	12 th -June	Fixed Bug 99
FileB	revision 1.5	by Alice at 12:35	12 th -June	Fixed Bug 99
FileC	revision 1.3	by Bob at 12:40	15 th -June	Fixed Bug 100



Looking at the above we can see the Alice's changes on the 12th-June are both for the same item and likely to be distinct from Bob's change on the 15th-June.

How do I use this feature?

For simple use you can rely on the default parameters of the tool and make use of the provided `-average` and `-predict` options without worrying about change grouping.

For greater accuracy you may wish to tweak the identification process using the following options.

`-identifychange groups` tells the tool that you require it to try and identify change groups automatically. (this is the default when `-average` or `-predict` is specified)

`-setdatethreshold` sets the time period between which two files by the same author are assumed to be part of the same change group.

`-setcommentdatethreshold` sets the time period between which two files having the same change comment are assumed to be part of the same change group.

You may want to set this value to more than the normal date threshold as you can be more confident that file deltas with the same comment belong together.

It may occur that parts of a change are booked in to the version control system by different people in this case you can specify `-ignoreauthor`. Otherwise the tool assumes that changes by different authors (and that have different change comments) are distinct.

Storing and overriding change groups

The change groups identified automatically by the software are only best guesses.

Your software process may be such that you have more detailed information available elsewhere.

If this is the case you can import it using the `-importchange groups` option.

Change groups identified by the tool can also be exported using the `-exportchange groups` option

E.g. if you wish to compare the automated analysis with your own records.

If your own records are incomplete you can combine the import of change groups with the automated analysis and export the results to help fill in the blanks.

The tool supports a simple text format for importing and exporting change groups.

This is described in the appendix.

Command Line Options Summary

Option	Parameters	Purpose
<code>-help</code>	None	Show basic usage information
<code>-version</code>	None	Show version details for the tool
<code>-database</code>	<database file name>.	Specify a database to use – if necessary creating it. Defaults to <code>srcmetrix.db</code>
<code>-project</code>	<project name>	Specify a project name. Defaults to "unknown"
<code>-listprojects</code>	None	Lists the projects stored in the database
<code>-nocommit</code>	None	Do not commit changes to the database.
<code>-changes</code>	None	Report change information only.
<code>-summary</code>	None	Report summary information only



-reportall	None	Report metrics for unmodified files.
-report	<report file name>	Generate report to the given file.
-history	None	Analyse file history (requires version control system)
-averagechange	None	Analyse project history and generate a characteristic change
-predict	<number of changes>	Estimate the time and effort required to make changes to the project.
-vcs	[CVS/PVCS]	Specify the version control system to use. By default the tool attempts to guess using available information.
-repository	<repository name>	Specify the version control repository to be used for historical analysis
-showlicense	None	Show registration details associated with the license including how many days remaining.

Change Grouping Options

Option	Parameters	Purpose
-identifychange groups	None	Analyse version control archives and combine changes to individual files into groups
-importchange groups	<change groups file>	Import change groups from the specified file
-exportchange groups	<change groups file>	Export identified change groups to the specified file
-ignoreauthor	None	Ignore the author field when combining file changes
-setdatethreshold	<number>(h/m/s)	Set the limit of difference in time between changes to files after which they will be considered part of distinct change groups. (note: does not apply to changes identified by comments)
-setcommentdatethreshold	<number>(h/m/s)	Set the limit of difference in time between changes to files for which the same comment has been used after which they will be considered part of distinct change groups.

Performance

First use

When first run Source Metrix has to analyse your project in detail. This can be time consuming particularly if any of the historical analysis & prediction features are used. The results are stored in a database so that subsequent runs are much more efficient.

Oversight Thread

Source Metrix contains a special oversight system that is used to monitor its own performance.

If processing is taking a long time a timeout warning may be issued. This is more common on slower system.

In extreme circumstances the program may even restart itself to help ensure optimal performance.

Network Databases

Source Metrix can work with databases on network drives. However, in this circumstance it is important to ensure multiple users do not access the database simultaneous or the results could be unpredictable.

Future Enhancements

Speeding up the initial parsing process is a priority for future releases.

Appendix A: Change Grouping Text Format

This appendix describes the change group description format:

The file should consist of a list of changes at minimum consisting of the list of file names changed and the associated version numbers.

Each such change should be bracketed within a "ChgID:" line containing an optional change identifier and "EndChg" on a line by itself

E.g.

```
ChgID: Change 1
-r1.1 FileA.cpp
-r1.2 FileA.h
EndChg
```

The tool also supports a number of optional fields for additional information. Each such field is indicated by its name followed by a colon and a space then the data for that field.

E.g.

```
Author: Alice
```

The description field is the only field that may span multiple lines.

It must be terminated with "EndDesc" on a line by itself.

```
Description: These files were changed by Alice to fix BugID 99
EndDesc
```

Example:

```
ChgID: Change 1
RequestID: BugID 99
Description: These files were changed by Alice to fix BugID 99
EndDesc
Author: Alice
Integrator: Alice
DateIntegrated: 12-June-2005
-r1.1 FileA.cpp
-r1.2 FileA.h
EndChg
```

You may also include comments preceded by a // (i.e. C++ style). Please note that these will not be stored or re-exported by the tool.

The complete list of permitted fields is tabulated below:

Field	Purpose
ChgID:	Identify a new change group
Description	Begin a description for the change
Author:	Who wrote the change
Integrator:	Who integrated into the version control system (may be same or different from



	author depending on your process)
RequestID:	Identifier associated with the change request.
DateIntegrated:	The date the change was integrated

Legal Blurb

This software and all materials are copyright Bruce S. O. Adams 2005-2006 – All rights reserved.

The software comes with no warranty either express or implied. The provider is not responsible for any damage that may occur either directly or indirectly as a result of using this product. You do so at your own risk.

You may not disassemble, hack or otherwise reverse engineer this software without express permission.

This software and materials may be freely distributed provided all materials are enclosed and that these statements are adhered to.

You are licensed to use this software only as agreed by the provider. That is for a period of 30 days evaluation or for the period of a lease purchased from the provider.

If you have purchased a lease you may use it on one machine at a time only. Non evaluation licenses may not be distributed or used on multiple machines without express permission from the provider.

By using this software you are implicitly agreeing to these terms.