

**Essential**

**Project Manager (EPM)**

***ADA, Assembly, ASP, C#, C/C++, CSS, Fortran, IDL, HTML, Java, JavaScript, JSP, MMP, Perl, PHP, PL/SQL, PowerBuilder, Python, Ruby, ShellScript, Textfiles, UCode, VB6 / VB.NET / VBScript, VHDL, Windows Batch and XML***

User Guide

*2.3.0.0*

# Contents

[Contents 2](#_Toc367094675)

[Document 2](#_Toc367094676)

[Purpose 2](#_Toc367094677)

[Readership 2](#_Toc367094678)

[Version History 3](#_Toc367094679)

[EPM Filelists 6](#_Toc367094680)

[Standard EPM Project file 6](#_Toc367094681)

[EPM Project file referencing Microsoft DSP 6](#_Toc367094682)

[Installing MySQL (optional) 7](#_Toc367094683)

[Installing Essential Project Manager on Microsoft Windows Vista/7/8 7](#_Toc367094684)

[Running Essential PM 8](#_Toc367094685)

[Installing a License 8](#_Toc367094686)

[Command Line options 8](#_Toc367094687)

[Combined PM Generation and Report 9](#_Toc367094688)

[Two-stage Single Project Generation and Report 9](#_Toc367094689)

[FAQ 10](#_Toc367094690)

[How does EPM know which parser to use? 10](#_Toc367094691)

[How can I limit the metrics that are output by the reports? 12](#_Toc367094692)

[What does line\_type in CSV reports mean? 13](#_Toc367094693)

[Why do I get ‘You do not have a valid license’? 13](#_Toc367094694)

[What does the error stating the MySQL cannot start mean? 13](#_Toc367094695)

[What do the Metric IDs and Codes mean? 14](#_Toc367094696)

# Document

## Purpose

This document is a User Guide for the Essential Metrics PM command-line source code metrics tool.

## Readership

This User Guide is intended for end-users, system administrators. Management and non-technical users should refer to our website – <http://www.powersoftware.com/epm/>

## Version History

|  |  |  |
| --- | --- | --- |
| 2.1.0.0 | 22-Jun-2011 | **10963** : Added new UCode language support  **10964** : Added new Modular Maths Processer language support |
| 2.0.0.0 | 8-Oct-2010 | **10648** : Implement PowerHash for Operand Detection  **10527** : Correct typo in default EPM.XML |
| 1.22.0.0 | 16-Mar-2010 | **10609** : Under the hood performance improvements |
| 1.21.1.0 | 5-Mar-2010 | **10576** : Calculate Changed/Unchanged within core EPM  **10579** : Add Metric Type (General, Churn) to metric table |
| 1.21.0.0 | 17-Feb-2010 | **10564** : x64 Compatibility  **10556** : CSV - deleted files have 0 CRN\*LOC values  **10570** : Start EPMdb automatically on installation, stop prior to removal  **10567** : Do not attempt to start/stop the MySQL database - it is started automatically from Services  **10568** : Upgrade MySQL database to 5.1 |
| 1.20.0.0 | 12-Dec-2009 | **10508** : Added PowerBuilder (PB) Language to EPM |
| 1.19.4.0 | 17-Nov-2009 | **10501** : Removed extraneous debug output in Diff causing occasional hang |
| 1.19.3.0 | 27-Oct-2009 | **10499** : Ensure integrity of build (following missing files in 1.19.2.0). |
| 1.19.2.0 | 15-Sep-2009 | **10479** : Add PLOC (Preprocessor LOC) for Preprocessor Directives to C/C++ and Assembly  **10483** : Add logging to EPM  **10488** : File extensions that do not exist in EPM.XML are excluded |
| 1.19.0.0 | 22-Aug-2009 | **10450** : Added CSS Stylesheet (SS) Language to EPM  **10452** : Added Ruby (RB) Language to EPM  **10455** : Added Windows Batch File (WB) Language to EPM  **10456** : Corrected issue with Halstead Bug Prediction (B) rounding to 0 or 1. |
| 1.18.0.0 | 27-Jul-2009 | **10454** : Add Text files (TX) Language to EPM  **10453** : Add Shell Script (SH) Language to EPM  **10457** : Investigate Intel Issue: EPM crash parsing 'MKL sources'  **10451** : Add Fortran (FT) Language to EPM |
| 1.17.3.0 | 12-May-2009 | **10450** : Ensure commas in filenames do not disrupt CSV reports. |
| 1.17.2.0 | 05-May-2009 | **10441** : Ensure free text within HTML is counted as SLOC. |
| 1.17.1.0 | 30-Apr-2009 | **10436** : Allow setting of language for empty file extension using space (" ") |
| 1.17.000 | 18-Apr-2009 | **10400** : Added JavaScript Language (JT)  **10401** : Added HTML Language (HT)  **10402** : Added Assembler Language (AY)  **10403** : Added Python Language (PY)  **10415** : Perl parser optimisation  **10416** : C# parser optimisation  **10418** : XML parser optimisation  **10420** : Ada parser optimisation  **10421** : ASP parser optimisation  **10422** : C/C++ parser optimisation  **10424** : Java parser optimisation  **10426** : PHP parser optimisation  **10428** : VB parser optimisation  **10430** : Add Churn metrics (CRN\_SLOC, CRN\_LLOC & CRN\_FILE) |
| 1.16.005 | 30-May-2008 | **366** : Ensure EPM has the current MySQL 5.0+ drivers. |
| 1.16.004 | 26-Apr-2008 | **360** : Build issues caused by compilation under VS2008 (upgraded from VC++6). |
| 1.16.003 | 24-Apr-2008 | **359** : LLOC whitespace causing erroneous Diff values  **355** : LLOC mismatch – C+-Style comments within C-style comments  **356** : DEL\_LLOC not being output to CSV file |
| 1.16.002 | 04-Jan-2008 | **330** : Improved links in footer on HTML (h2) report. |
| 1.16.001 | 14-Dec-2007 |  |
| 1.16.000 | 11-Dec-2007 | **325** : Added JSP and XML Parsing. |
| 1.15.004 | 28-Nov-2007 | **323** : Resolved issues with missing PHP metrics. |
| 1.15.003 | 01-Nov-2007 | **321** : Ensure Logical Lines are calculated corrected in Ada, ASP & C/C++/Java. |
| 1.15.002 | 26-Oct-2007 | **319** : Remove duplicate files in projects before analysing |
| 1.15.001 | 28-Jul-2007 | **317** : Changed default VHDL extension to \*.vdh |
| 1.15.000 | 26-Jul-2007 | **316** : Added VHDL Parsing |
| 1.14.000 | 10-Jul-2007 | **314** : Added IDL Parsing |
| 1.13.000 | 10-Nov-2006 | **301** : Added PHP Parsing |
| 1.12.000 | 27-Oct-2006 | **117** : Added ASP Parsing |
| 1.11.000 | 18-Jul-2006 | **272** : Added Changed Logical Lines of Code (CHG\_LLOC, DEL\_LLOC, ADD\_LLOC) |
| 1.10.004 | 06-Jun-2006 | **255** : CSV Reporting correctly shows DEL\_SLOC and DEL\_FILE  **257** : Metrics Sets work for XML and CSV Reports |
| 1.10.003 | 02-Jun-2006 | **244** : Resolved System Error 3 (MySQL service issues)  **252** : Add Min/Max/Avg to Project window for Halstead metrics |
| 1.10.002 | 11-May-2006 | **241** : Ensure EPM can be executed from any directory |
| 1.10.001 | 10-Apr-2006 | **230** : Ensure Ada parsing is case-insensitive |
| 1.10.000 | 26-Mar-2006 | **218** : Ignore Halstead metrics at the Project level  **220** : Metrics for New files are all Zero  **221** : CSV and XML Reports not obeying the specific metric visibility rules |
| 1.09.003 | 18-Mar-2006 | **212** : Employ the Diff specific to each Language to ensure correct Changed metrics |
| 1.09.000 | 21-Feb-2006 | **188** : DSP Project file parsing added |
| 1.08.002 | 14-Feb-2006 | **187** : Various MySQL and EPM bugs quashed |
| 1.08.000 | 28-Jan-2006 | **168** : Added Metric Sets to the EPM XML Configuration file |
| 1.07.000 | 20-Dec-2005 | **160** : Use Windows Services rather standalone mode for MySQL  **164** : Added NFILE metric |
| 1.06.000 | 23-Aug-2005 | Added XML Configuration |
| 1.05.000 | 15-Aug-2005 | Added Perl Parsing |
| 1.04.000 | 02-Aug-2005 | Added ADA Parsing |
| 1.03.000 | 19-May-2005 | Added PL/SQL Parsing |
| 1.02.000 | 13-May-2005 | Added VB6 and VB.NET Parsing |
| 1.01.000 | 9-May-2005 | Added Java Parsing |
| 1.00.015 | 4-May-2005 | Added C/C++ Parsing |
| 1.00.014 | 29-Mar-2005 | Added XML reporting. |
| 1.00.013 | 27-Mar-2005 | Added CSV reporting. |
| 1.00.012 | 25-Mar-2005 | Corrected detection of changes from non-zero Difference in standard metric values to non-zero CHG\_SLOC, ADD\_SLOC, DEL\_SLOC.  To prevent loss of precision due to large Project E values, represent E in thousands (k). |
| 1.00.011 | 20-Mar-2005 | Corrected rounding issues and case-insensivity with filenames. |

# EPM Filelists

Essential Project Manager accepts “Project files”. A Project file is required to describe the project’s key attributes: its name, a snapshot date, as well as the files to be parsed within the project.

The Standard EPM Project contains the name of each file to be parsed within the project file itself. It is also possible to create an EPM Project that references an external project file (currently Microsoft DSP files are supported) to minimize the amount of additional configuration required to integrate EPM into the build process.

## Standard EPM Project file

A Standard EPM Project file is structured as follows:

Line 1: **Project Name**

Line 2: **Project Snapshot date** (a text representation of the date snapshot taken)

Line 3: **Base directory** (useful for cropping long paths and essential for PM comparisons)

Line 4-end: full path and filename of each file

You can create a filelist using a DOS Batch file similar to the following:

@echo off

set FILELIST=filelist.txt

set PATH=c:\tmp\c#code\mono\mono-1.0.5

echo Mono Version 1.0.5 > %FILELIST%

echo 03-Feb-2005 >> %FILELIST%

echo %PATH% >> %FILELIST%

dir %PATH%\\*.cs /s /b >> %FILELIST%

## EPM Project file referencing Microsoft DSP

To reference your Microsoft DSP project file within an EPM Project, structure your file as follows:

Line 1: **Project Name**

Line 2: **Project Snapshot date** (a text representation of the date snapshot taken)

Line 3: @DSP=<your DSP file>

For example:

My New Project

21/02/2006

@DSP=c:\Dev\Project1\Project1.dsp

Note that there is no “Base Directory” specified. This is because EPM intelligently derives the base directory by extracting the most common part of the path from the absolute pathname of every file referenced in the DSP file. This enables EPM to compare different versions of a Microsoft VS project without requiring a configuration headache.

# Installing MySQL (optional)

If you wish to use an existing MySQL server in the organization, you can do so using the –s (and –u/-p if required) parameter.

Otherwise, a local MySQL database will be started implicitly as Essential Metrics runs.

## Installing Essential Project Manager on Microsoft Windows Vista/7/8

Windows’ User Account Control prevents EPM from installing and starting the MySQL service, unless you right-click on the Setup.exe and choose **Run as Administrator**.

# Running Essential PM

## Installing a License

You must obtain either a time-locked (for testing) or node-locked/floating (following purchase) license from Power Software. The recommended way to install the license is to place it where you installed Essential Metrics (e.g. C:\Program Files\Power Software\EPM) and then simply to call ‘epm’ at the command line. This will prompt for you to Browse to the license.dat file.

## Command Line options

Usage: epm options name

**Options**

-s MySQL server Name of the MySQL server to store data

-u MySQL user User to connect to MySQL server (**must accompany –p**)

-p MySQL pass Password to connect to MySQL server (**must accompany –u**)

-fX filelist Obtain files from the given file list (f1=new, f2=old)

-m metrics\_set Name of the Metrics Set (defined in epm.xml) to apply to reports.

-c csv\_file Create a CSV report.

-h2 html\_dir Create multiple page HTML report.

-x xml\_file Create an XML report.

-l log\_file Log output of the session to a logfile.

-? Show this help screen.

**Name** A name to give the project.

You can either operate as a two-stage process:

1. Generate metrics
2. Output reports

Or perform both tasks together. You must specify one or other or both, you cannot omit both the filelist and reporting options – Essential Metrics would have nothing to do.

## Combined PM Generation and Report

epm -f1 C:\tmp\c#code\mono\mn\_new.txt

-f2 C:\tmp\c#code\mono\mn\_old.txt -h2 html -s knox mono1

Explanation:

-f1 is the first project filelist, considered to be the new project

-f2 is the old project filelist, or baseline, against which the comparison will be performed

-h2 specifies an HTML report to the ‘**html**’ directory (if this does not exist, you will be prompted for creation)

-s specifies ‘**knox**’ as the MySQL server

mono1 is the MySQL database that will be created to store the results.

**WARNING: the database will be dropped and recreated each time –fX options are passed to Essential Metrics – ensure you do not use the name of a existing database if using a separate MySQL Server.**

## Two-stage Single Project Generation and Report

epm -f1 C:\tmp\c#code\mono\mn\_new.txt -s knox mono2

epm -h2 html -s knox mono2

If you wish to copy the ‘style.css’ file included with the build to any HTML report directories you create, you will see the output is formatted colorfully. You may also amend the style.css to match any corporate/departmental standards before publishing to an intranet.

# FAQ

## How does EPM know which parser to use?

EPM is supplied with a default XML Configuration file that instructs the tool to apply different languages parsers to the files based on the following file extensions:

|  |  |  |
| --- | --- | --- |
| **Language** | **Extensions** | **Parser Code** |
| ADA | .a .ada .adb .ads | AD |
| ASP | .asp .aspx | AS |
| Assembly | .asm | AY |
| C++ | .cpp .c .hpp .h | CP |
| C# | .cs | CS |
| CSS | .css | SS |
| Fortran | .f .f90 | FT |
| IDL | .idl | ID |
| HTML | .htm .html .htp | HT |
| Java | .java | JV |
| JavaScript | .js | JT |
| JSP | .jsp | JS |
| Modular Maths Processer | .mmp | MP |
| PHP | .php | PH |
| Perl | .pl | PL |
| PL/SQL | .sql .ora | S1 |
| PowerBuilder | .srd .srf .srs .sru .srw | PB |
| Python | .py | PY |
| Ruby | .rb | RB |
| Shell Script | .ash .bash .bsh .csh .sh .tcsh .tsh .zsh | SH |
| Textfile | .cvs .install .readme .tsv .txt | TX |
| UCode | .uc | UC |
| VB.NET | .vb | VB |
| VB6 | .frm .bas .cls | VB |
| VBScript | .vbs | VB |
| VHDL | .vhd | VH |
| Windows Batch | .bat .cmd | WB |
| XML | .xml .xsd .xsl .xslt .wsml | XM |

**You cannot simply add a language of your own**. EPM only understands the **Parser Codes** listed above. Should you need a new language not listed here, please contact us to discuss.

*If* you need to modify this configuration, you may edit the EPM.XML configuration file supplied with the tool, but please **take a backup before you start** and if you experience any errors and you report these, please **include your EPM.XML file** and state what you were trying to achieve.

The default file looks like:

|  |  |
| --- | --- |
| <?xml version="1.0" encoding="UTF-8"?>  <epm>  <lang name="C++" parser="CP">  <filetype name="Source">  <ext name="Class" value="cpp" />  <ext name="C File" value="c" />  </filetype>  <filetype name="Header">  <ext name="C++ Header" value="hpp" />  <ext name="C Header" value="h" />  </filetype>  </lang>  <lang name="C#" parser="CS">  <filetype name="Source">  <ext name="Class" value="cs" />  </filetype>  </lang>  …  </epm> | This extract from the file shows the definitions for two Languages: C++ and C#.  Within each lang entity there are filetype entities. **The C++** Language has two filetypes: **Source** and **Header**. Each filetype then has one or more ext entities representing the file extensions that comprise the filetype.  **C#** only has one filetype – **Source** – and a single extension – **cs**.  *Extensions are the characters after the last period (.) in the filename.* |

To add new filetype or ext entities, simply edit the file using a text editor, copy/paste an existing block and modify the name and value attributes accordingly.

The values in the **name** attributes are free-text. In a future version of EPM it will be possible to report on these values, including viewing files/metrics grouped by the same filetype (e.g. Source) across several languages.

If you have files that have no extension, you may use a value of “ “ (single space). This means that **all** files with no extension will be parsed using the language parser you have selected.

<ext name="MyFileType" value=" " />

The **parser** attribute of the **lang** entity must be one of the Parser Codes listed in the first table.

## How can I limit the metrics that are output by the reports?

You can limit the metrics that are output to the EPM reports by creating “Metric Sets” in the epm.xml Configuration file.

A Metric Set is a grouping of Metrics identified by a name. You can then pass that *name* into EPM via the “-m *name*” option (see **Command Line options**). An example follows:

|  |  |
| --- | --- |
| <?xml version="1.0" encoding="UTF-8"?>  <epm>  <lang name="C#" parser="CS">  <filetype name="Source">  <ext name="Class" value="cs" />  </filetype>  </lang>  <sets>  <set name="Core Metrics">  <met id="100" upper="10000" />  <met id="101" />  <met id="102" />  <met id="109" upper="10000" />  <met id="110" lower="0" upper="100" />  <met id="111" lower="100" upper="1000000" />  <met id="112" upper="10" />  <met id="116" />  <met id="118" />  <met id="119" />  <met id="121" />  <met id="122" />  </set>  <set name="Code Lines">  <met id="100" />  <met id="101" />  <met id="119" />  <met id="120" />  <met id="121" />  </set>  </sets>  </epm> | After the “lang” entities, you would create a **<sets>** entity.  Within this, you can create one or more **<set>** entities.  The example opposite has two Metric Sets defined: “Core Metrics” and “Code Lines”. Be sure to give the set a name, so that you can refer to it when you call EPM. |

Within each set, you list the Metrics (by ID – see **What do the Metric IDs and Codes mean?**) and you can optionally put lower and upper limits on these metrics. Future EPM functionality will allow “Violating Metrics” reports to be produced showing those files whose metrics are above/below the limits specified.

## What does line\_type in CSV reports mean?

|  |  |  |
| --- | --- | --- |
| **Line Type** | **Description** | **Meaning** |
| **C** | Changed | The File/Project has Changed. This line gives details of the New File/Project and the original metrics can be found on a corresponding Old (**O**) line. |
| **O** | Old | Where a File/Project has Changed, the Old line will also be output to the CSV file. |
| **X** | Diff | Where a File/Project has Changed, the Diff line will also be output to the CSV file, showing the difference between Changed (**C**) and Old (**O**) lines. |
| **U** | Unchanged | The File/Project has not changed. |
| **N** | New | The File is New, it has been added to the Project. |

Deleted files are not shown in CSV reports.

## Why do I get ‘You do not have a valid license’?

If you have not requested a license from [laurence.arthur@powersoftware.com](mailto:laurence.arthur@powersoftware.com), please do so. Once you have received this file (supplied as an attachment called ‘license.dat’), please detach this to the folder in which you installed EPM, by default:

C:\Program Files\Power Software\Essential Metrics PM

If there is an existing license.dat, overwrite it. Then try re-executing EPM.exe. If you continue to experience problems, send an email [support@powersoftware.com](mailto:support@powersoftware.com) with the License [Code] value you see on starting up EPM.

## What does the error stating the MySQL cannot start mean?

If you are relying on EPM to start a MySQL database for you, please ensure that you wait a few seconds between running successive EPM sessions. MySQL takes a moment or so to start and stop the service at the beginning and end of an EPM session.

## What do the Metric IDs and Codes mean?

We use a unique numeric code for each metric, as well as an alpha code which is more descriptive. For efficiency reasons, these numeric codes may change between releases, when a new metric is inserted into the list. We upgrading, be sure to check your Metric Sets and any other interfaces that rely upon the Metric ID.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Code** | **Description** | **Project** | **File** |
| 100 | LOC | Lines of Code | ✓ | ✓ |
| 101 | SLOC | Source Lines of Code | ✓ | ✓ |
| 102 | SLOC\_NAT | Source Native Lines of Code | ✓ | ✓ |
| 103 | SLOC\_TAG | Source Tag Lines of Code | ✓ | ✓ |
| 104 | SLOC\_HTM | Source HTML Lines of Code | ✓ | ✓ |
| 105 | SLOC\_SCR | Source Script Lines of Code | ✓ | ✓ |
| 106 | PLOC | Preprocessor Directive Lines of Code | ✓ | ✓ |
| 107 | LLOC | Logical Lines of Code (semi-colon count – formerly NSC) | ✓ | ✓ |
| 108 | N1 | Total No. of Operators |  | ✓ |
| 109 | N2 | Total No. of Operands |  | ✓ |
| 110 | n1 | No. of unique or distinct Operators |  | ✓ |
| 111 | n2 | No. of unique or distinct Operands |  | ✓ |
| 112 | N | Halstead program Length (calculated as N1 + N2) |  | ✓ |
| 113 | n | Halstead program Vocabulary (calculated as n1 + n2) |  | ✓ |
| 114 | V | Halstead Volume (calculated as V = Nlog2n) |  | ✓ |
| 115 | D | Halstead program Difficulty |  | ✓ |
| 116 | E | Halstead program Effort (calculated as D \* V) |  | ✓ |
| 117 | B | Halstead Bug Prediction |  | ✓ |
| 118 | J\_COM | Java-style Comment Lines | ✓ | ✓ |
| 119 | C\_COM | C-style Comment Lines | ✓ | ✓ |
| 120 | EOL\_COM | To End of Line Comment Lines | ✓ | ✓ |
| 121 | COM\_LOC | Total Comment Lines | ✓ | ✓ |
| 122 | BYTES | File size in bytes | ✓ | ✓ |
| 123 | NFILE | Number of Files | ✓ |  |
| 124 | CHG\_SLOC | Changed Source Lines of Code | ✓ | ✓ |
| 125 | DEL\_SLOC | Deleted Source Lines of Code | ✓ | ✓ |
| 126 | ADD\_SLOC | Added Source Lines of Code | ✓ | ✓ |
| 127 | CRN\_SLOC | Churn Source Lines of Code (CHG\_SLOC + DEL\_SLOC + ADD\_SLOC) | ✓ | ✓ |
| 128 | CHG\_LLOC | Changed Logical Lines of Code | ✓ | ✓ |
| 129 | DEL\_LLOC | Deleted Logical Lines of Code | ✓ | ✓ |
| 130 | ADD\_LLOC | Added Logical Lines of Code | ✓ | ✓ |
| 131 | CRN\_LLOC | Churn Logical Lines of Code (CHG\_LLOC + DEL\_LLOC + ADD\_LLOC) | ✓ | ✓ |
| 132 | CHG\_FILE | Changed Files | ✓ |  |
| 133 | DEL\_FILE | Deleted Files | ✓ |  |
| 134 | ADD\_FILE | Added Files | ✓ |  |
| 135 | CRN\_FILE | Churn Files (CHG\_FILE + DEL\_FILE + ADD\_FILE) | ✓ |  |