



**Free/Libre and Open
Source Software Metrics**



Sponsored through Framework Programme Sixth (Call 5) by

The FLOSSMetrics Consortium consists of: Universidad Rey Juan Carlos, University of Maastrich, Wirtschaftsuniversitaet Wien, Aristotle University of Thessaloniki, Conecta s.r.l., Zea Partners and Philips Medical Systems PMS Nederland B.V.

Document Information

Version: 1.1
Date : Oct 02, 2008
 revision: 1

Owning Partner:
 WUW

Author(s):
 Santiago Dueñas
 Carlos García Campos

Reviewer(s):
 Jesus M. Gonzalez-Barahona

To:
 PUBLIC

Purpose of distribution:
 Final version

**Printed
 on at**

Status:

- Draft
- To be reviewed
- Proposal
- Final/Released

Confidentiality:

- Public - Intended for public use
- Restricted - Intended for FLOSSMetrics consortium only
- Confidential - Intended for individual partner only

Deliverable ID: D2.2

Title:

Implementation of Retrieval System

License for distribution of this report:


This work is licensed under a [Creative Commons Attribution-Share Alike 2.5 License](http://creativecommons.org/licenses/by-sa/2.5/).
 (The license can be found in <http://creativecommons.org/licenses/by-sa/2.5/>)

The original version of this document is available at <http://flossmetrics.org>

License for the attached software:

This software is licensed under the GNU GPL version 2.0 or later license, except when otherwise stated in the source code of the system. See the source code for more details.

(The license can be found in <http://fsf.org>)


	Implementation of Retrieval System Deliverable ID: D2.2	Page: 2 of 12
		Version: 1.1 Date: Oct 02, 08
		Status: Final Confid: Public

Deliverable: D2.2

Title: Implementation of Retrieval System

Executive Summary:

The objective of this report is to provide information about Retrieval System of the FLOSSMetrics project. Its main target is to retrieve information from public repositories used by libre software projects, and feed that information to the main database of the project, where those data will be analysed and organised in several ways. This deliverable includes the source code of the system, and its installation and user guides.

	<p>Implementation of Retrieval System</p> <p>Deliverable ID: D2.2</p>	Page: 3 of 12
		Version: 1.1 Date: Oct 02, 08
		Status: Final Confid: Public

CHANGE LOG

Ver.	Date	Author	Description
0.1	01/08/2007	Santiago Dueñas	Initial version
1.0	06/09/2007	Jesus M. Gonzalez-Barahona	Final version
1.1	02/10/2008	Carlos García Campos	Add adapters information

APPLICABLE DOCUMENT LIST

Ref.	Title, author, source, date, status	Deliverable Identification



	<p>Implementation of Retrieval System</p> <p>Deliverable ID: D2.2</p>	Page: 4 of 12
		Version: 1.1 Date: Oct 02, 08
		Status: Final Confid: Public

TABLE OF CONTENTS

1. Introduction	5
2. Implementation Details	6
2.1 Retrieval System Core	6
2.2 External Tools Adapters	7
3. Installation	9
3.1 Availability	9
3.2 Previous Requirements	9
3.3 Installing from sources	9
4. User Guide	11

	<p>Implementation of Retrieval System</p> <p>Deliverable ID: D2.2</p>	Page: 5 of 12
		Version: 1.1 Date: Oct 02, 08
		Status: Final Confid: Public

1. INTRODUCTION

The Retrieval System is a software package that automates the retrieval (and partially, analysis) of data from public repositories about libre (free, open source) software development. It actually is mainly a front-end that organises and schedules the execution of a set of third party retrieval and analysis tools.


The system receives as input from information about which projects will be retrieved and analysed. With the given data, the system search for the datasource repositories. Then, it downloads the data and then executes the available tools over them. Once the work had been finished, the results are stored into a database.

This system is one of the main parts of the FLOSSMetrics project, and has a target the retrieval and analysis of information from the repositories of thousands of projects. At the moment of writing this report, it has already been tested with a list of about 100 different projects, and is in production at the premises of the URJC.

The design of the Retrieval System was described in the the “D2.1 – Design of Retrieval System” document. For this reason, this deliverable only includes the technical decisions taken during the developing of the system and the source code of the system.

The structure of this document is as follows. The report starts with a list of the implementation details such as used technologies during the developing of the system. Then, the installation manual, including the links to the source code. The last part contains the user guide.

This report describes the current implementation of the Retrieval System. However, as the project evolves, the system will be improved and augmented in functionality, mainly by adding new modules to its plugable architecture.

	<p>Implementation of Retrieval System</p> <p>Deliverable ID: D2.2</p>	Page: 6 of 12
		Version: 1.1 Date: Oct 02, 08
		Status: Final Confid: Public


2. IMPLEMENTATION DETAILS

2.1 RETRIEVAL SYSTEM CORE

The Retrieval System has been implemented according to the design features described into the “D2.1 – Design of Retrieval System” document. This section contains the general decisions taken during developing process.

- **Programming Language:** The programming language selected to implement the system has been Python. Most of the well-known retrieval and analysis tools are written in this language, thus, this increases the reuse of code and the interoperability. The repository finder is also programmed in Python.
- **Database Management:** MySQL has been selected to store the information related with the retrieval system. In next versions other database management systems will be supported. To facilitate the interaction with the database the STORM library, an ORM (Object Relational Mapping), will be used. It converts database rows into Python objects.
- **Supported Tools:** For this version, three retrieval and analysis tools are already supported: CVSanaly, MalingListStats and Sloccount. Probably these are three of the best existing applications that allow to analyse two kinds of repositories: SCMs and mailing lists. Next versions of the Retrieval System will include support for further analysis tools.
- **General Log:** All the system activity is stored into a file. This gives some advantages over the use of a database. For example, the I/O operations and the visualisation of the information are faster and easier than in a database. The log will contain data related to the produced events such as the date and which component raise it. And example of the stored information is the next:

```
[2007-09-06 19:36:45.536877 - TaskManager] Info: Task with id: 2 selected
[2007-09-06 19:36:45.536877 - TaskManager] Info: Task with id: 2 selected
[2007-09-06 19:36:45.538820 - TaskManager] Info: Executing task
[2007-09-06 19:36:45.571228 - CVSanaly] Info: Adapter loaded and generating command
[2007-09-06 19:36:45.571325 - CVSanaly] Info: Executing command
[2007-09-06 19:36:45.596086 - CVSanaly] Error: Code None: [Errno 2] No such file or directory
[2007-09-06 19:36:45.596214 - TaskManager] Error: Task failed. [Errno 2] No such file or directory
[2007-09-06 19:36:45.599608 - TaskManager] Info: Looking for tasks to execute
```

	<p>Implementation of Retrieval System</p> <p>Deliverable ID: D2.2</p>	Page: 7 of 12
		Version: 1.1 Date: Oct 02, 08
		Status: Final Confid: Public


- **Datasources Downloader:** The API provided by the Repository Handler¹ library is used to implement the access and download processes of the different datasources, therefore not reinventing a new API. The Repository Handler works with several datasources types such as SCMs and mailing lists. It is programming in Python and developed by of the FLOSSMetrics partners: GsyC/Libresoft (URJC).
- **Octopus Interaction:** The interaction with the repository finder (Octopus) is performed using the API provided by this system.
- **Input Data:** The system provides a SOAP interface that allows to insert new projects to retrieve and analyse. The input will consist on a XML stream that will include the forges that store the projects, the projects to retrieve, and the tools that should be executed over the projects' data.
- **Network Services Framework:** The facilities of Twisted have been used as a network services framework (something needed because the system will provide services for allowing remote interaction over a network). Twisted is a network framework written in Python that provides components to integrate network services, in a fast way, into other applications. In this version, only the SOAP service is supported, but in a future more network services could be added.

2.2 EXTERNAL TOOLS ADAPTERS


The Retrieval System supports external tools that perform the analysis over the datasources. At the moment there are four external tools supported: CVSAnalY2, MalingListStats, Bicho and Sloccount. The Retrieval System is designed to easily allow adding support for new external tools, based on a plugins system. Every external tool supported by the Retrieval System is wrapped by a plugin called Adapter.

An Adapter is, therefore, a plugin that implements the code necessary to run and watch an external tool. The Adapter knows how to run the tool and it also watches it in order to make sure the tool is running correctly. The output expected by the Retrieval System from an Adapter is a database. This means that if the external tool doesn't provide a database, the Adapter that wraps such tool must convert the output of the tool into a database. This is the case of the Sloccount Adapter.

¹ <https://forge.morfeo-project.org/plugins/scmsvn/viewcvs.php/utils/trunk/repositoryhandler/?root=libresoft-tools>

	Implementation of Retrieval System Deliverable ID: D2.2	Page: 8 of 12
		Version: 1.1 Date: Oct 02, 08
		Status: Final Confid: Public

In order to add support for new external tools, new adapters need to be written. Thanks to the plugin system this is the only thing needed, and no other part of the Retrieval System code needs to be changed.

	<p>Implementation of Retrieval System</p> <p>Deliverable ID: D2.2</p>	Page: 9 of 12
		Version: 1.1 Date: Oct 02, 08
		Status: Final Confid: Public

3. INSTALLATION

3.1 AVAILABILITY

The Retrieval System is free software distributed under the terms of the GNU GPL version 2 or any later version. It is hosted at the Morfeo Project forge where most of the GSyC/LibreSoft tools (such as CVSanaly2, MailingListStats or Bicho) and the Repository Finder are publicly available.

The current version of the Retrieval System 1.0, can be downloaded from the Morfeo Project forge in the form of a source code package at:

<https://forge.morfeo-project.org/frs/download.php/249/retrievalsystem-1.0.tar.bz2>

3.2 PREVIOUS REQUIREMENTS

Before to install the Retrieval System some libraries and applications must be available in the system.

The list is the next:


- storm (0.9)
- python-mysqldb (1.2.1)
- twisted (2.5)
- repositoryhandler (svn trunk – <https://forge.morfeo-project.org/projects/libresoft-tools/utills>)
- octopus (1.0 - svn trunk - <https://forge.morfeo-project.org/projects/libresoft-tools/octopus>)

3.3 INSTALLING FROM SOURCES

The installation process of Retrieval System is the one in use for most of the UNIX applications. First of all, download the latest version from https://forge.morfeo-project.org/frs/?group_id=33

Decompress the archive with the next command:

```
$ tar xvjf retrievalsystem-1.0.tar.bz2
```

	<p>Implementation of Retrieval System</p> <p>Deliverable ID: D2.2</p>	Page: 10 of 12
		Version: 1.1 Date: Oct 02, 08
		Status: Final Confid: Public

A new directory will be created under the current working directory. An INSTALL file is included in this directory with generic information about how to install it. The next step consists of running the configure script:

```
$ cd retrievalsystem-1.1
$ ./configure
```

The configure script can be run with different options in order to customize the installation, see INSTALL file or run `$. /configure --help` for further information. Once the configure script has finished without errors, the Retrieval System can be build just by typing make:


```
$ make
```

At the end of the build process, Retrieval System is ready to be used from the sources directory. Optionally it can be installed in the system, so that it can be used from any directory like any other UNIX command. In order to install Retrieval System an addition final step has to be taken²:

```
# make install
```

At this point, Retrieval System should be ready to be used like any other command available in the system.

² This last step has to be executed with root privileges in most of the cases, although it depends on the arguments used when running the configure script.

	<p>Implementation of Retrieval System</p> <p>Deliverable ID: D2.2</p>	Page: 11 of 12
		Version: 1.1 Date: Oct 02, 08
		Status: Final Confid: Public

4. USER GUIDE

Before to run the Retrieval System, the user must create the databases that will store the projects and the database for the retrieval system data.

The execution syntax of Retrieval System is simple:

```
Usage: retrieval_system [options]
```

A SOAP service is listening connections than will receive request for analyse projects. These request are XML streams that contain relevant information about the projects.


The structure of the XML is as follows:

- *projects*: its attributes are the URL and the type of the forge (sourceforge, gforge, an so on), in which the project is stored. This elements also should have a list of *project* elements.
- *project*: with the name and URL of the project to analyse and the wished priority of analysis. Also should have a list of *tools* elements
- *tools*: contains the tools to run over a project, depending on the type of the sources manipulated. The types can be: SCM for system control management, MLS for mailing lists, BTS for bug tracking systems and OTHERS for other types not defined yet.

In addition, this element has the *to_execute* attribute that will execute all the available tools of this type (ALL), only the tools of the list (SELECTION), or none (NONE).

- *tool*: contains the name o the tool and its version. Should have a list of *param* elements that establish the tool execution parameters.

And example of this XML could be the next:

	<p>Implementation of Retrieval System</p> <p>Deliverable ID: D2.2</p>	Page: 12 of 12
		Version: 1.1 Date: Oct 02, 08
		Status: Final Confid: Public

```

<projects url='https://forge.morfeo-project.org' type='gforge'>
  <project name='libresoft-tools'
    url='https://forge.morfeo-project.org/projects/libresoft-tools/'
    priority='2'>
    <tools type='SCM' to_execute='SELECTION'>
      <tool name='cvsanaly' version='1.0.1'>
        <param attr='db-user' value='root' />
        <param attr='db-password' value="" />
      </tool>
    </tools>
    <tools type='MLS' to_execute='ALL'>
      <tool name='mlstats' version='0.3'>
      </tool>
    </tools>
  </project>
</projects>

```