



Coverity, Inc.  
Professional Services

## ***Integration Scripts***

### ***Documentation***

January 19, 2011

## TABLE OF CONTENTS

---

Introduction .....	3
Development Workflow .....	3
Dependencies .....	4
Configuration Files.....	5
Scripts .....	8
1) test-scm-plugin.pl .....	8
2) assign-owner-to-unassigned-cids.pl .....	9
3) set-defect-severity.pl .....	10
4) email-notify-project.pl .....	11
5) email-notify-owners.pl.....	12
6) export-defect-handler.pl.....	13
About Coverity .....	14

## INTRODUCTION

---

These integration scripts are provided by the Coverity Professional Services group and are intended to support workflow integration.

A few important points regarding these scripts:

- They are provided as samples only and are *not* supported by Coverity Technical Support.
- There is no guarantee that they will work unchanged with future versions of Coverity.
- The integration scripts will almost certainly require source-code modification (in the form of plugins written in Perl) to get working.

That being said, the Professional Services group is committed to ensuring that these scripts are as high quality as possible and we highly value your feedback in being able to do so.

## DEVELOPMENT WORKFLOW

---

A typical development workflow integrated with Coverity (with the help of these scripts) might look like this:

1. A static analysis snapshot is generated and committed to a Defect Database.
2. New defects are assigned to owners based on an SCM lookup (*assign-owners-to-unassigned-cids*).
3. New defects are given an initial severity level based on checker type (*set-defect-severity*).
4. Project owners are alerted to new defects via email notification (*email-notify-project*).
5. Defect owners are reminded of recently assigned defects via email notification (*email-notify-owners*).
6. Review teams triage defects and defects to be fixed are manually promoted to an external issue tracking system via the Export button within CIM (*export-defect-handler*).

Note: the integration script that provides the corresponding functionality is listed in parenthesis.

## DEPENDENCIES

---

Scripts are installed to a particular location e.g. `/path/to/ps-scripts-5.3/scripts` but they can be relocated.

Scripts have the following dependencies:

- Network connection to the Coverity Integrity Manager
- Perl installation, version 5.8.1 or higher
- Coverity written Perl libraries
  - Installed under `/path/to/ps-scripts-5.3/Coverity-WS/lib`
  - Installed under `/path/to/ps-scripts-5.3/lib`
- Open source Perl libraries (available from <http://cpan.org>)
  - Class::Inspector
  - Log4Perl
  - SOAP::Lite - version 0.712 or higher
  - XML::Simple
  - XML::Parser
  - Other libraries may be required depending on the version of Perl installed

To test to see if all the libraries are installed correctly, invoke a script with the `--help` option. A usage summary should appear. If an error message appears instead then check your Perl installation for the necessary dependencies.

## CONFIGURATION FILES

---

Scripts rely on a XML format configuration file:

- /path/to/ps-scripts-5.3/scripts/coverity\_pse\_config.xml

The XML configuration file contains 3 main sections:

### 1) CIM information

```
<cim>
  <host>CIMHOST</host>
  <port>8080</port>
  <username>admin</username>
  <password>PASSWORD</password>
</cim>
```

The *host*, *port*, *username* and *password* fields should be filled in with the credentials for logging into the default CIM instance.

### 2) SCM and issue tracking system information

```
<systems>

  <system id="Accurev" type="scm">
    <plugin>Accurev</plugin>
    <repository></repository>
    <strip-path>/work/source/codebase123</strip-path>
    <basedir>/work/source/codebase123</basedir>
    <!--
    <username>script</username>
    <password>password</password>
    -->
  </system>
  ...
```

Each <system> has an *id* attribute and a *type* attribute. The *id* is an identifying label for a system and should be unique to that system. The valid *types* are:

- scm
- bts

```
<system id="scm-1" type="scm">
```

For both scm and bts (bug tracking system) types, a plugin must be identified e.g.:

```
<plugin>Subversion</plugin>
```

The remaining fields are passed in as key/value pairs into the plugin for processing. The fields and their purpose vary from plugin to plugin.

For example, the Subversion plugin expects the following:

```
<repository>file:///users/coverity/src/SVN/demo-code/trunk</repository>  
<strip-path>/users/coverity/src/sands/demo-code/</strip-path>
```

The repository element specifies the Subversion repository to connect to. The strip-path element normalizes the file path in which the Coverity defect was found in. This happens before querying the SCM system for owner information.

### 3) Mappings that associate CIM projects or streams to systems

```
<project-mappings>  
  <map>  
    <!-- pattern must be a perl regexp -->  
    <pattern>cms</pattern>  
    <systems>  
      <id>scm-1</id>  
    </systems>  
  </map>  
  <map>  
    <!-- pattern must be a perl regexp -->  
    <pattern>.*</pattern>  
    <systems>  
      <id>scm-1</id>  
      <id>bts-1</id>  
    </systems>  
  </map>  
</project-mappings>
```

Each <map> is comprised of a *pattern* and a series of *systems*.

The *pattern* tag contains a Perl regex expression that will be used to match on the name of a CIM project or stream. The “.” value is a wildcard that matches all names.

```
<map>
  <!-- pattern must be a perl regexp -->
  <pattern>.*</pattern>
  <systems>
    <id>scm-1</id>
    <id>bts-1</id>
  </systems>
</map>
```

The *systems* tag will contain a listing of *ids*. The *id* tag should contain the name of a scm or bts system. For each *pattern*, there may be multiple systems *but no more than one of each type*.

```
<map>
  <!-- pattern must be a perl regexp -->
  <pattern>.*</pattern>
  <systems>
    <id>scm-1</id>
    <id>bts-1</id>
  </systems>
</map>
```

The ability to include many mappings is necessary only when different projects or streams must map to different systems. In this situation the first mapping is checked for a match. If there is one, then its systems are used. Otherwise the scripts continue to check the second mapping for a match and so on. The final mapping should use the pattern “.” as a catch-all.

## SCRIPTS

---

Scripts are written in Perl and support a --help option for usage notes.

### 1) TEST-SCM-PLUGIN.PL

#### SYNOPSIS

```
test-scm-plugin.pl [options] --config CONFIG_FILE
--project PROJECT (OR --stream STREAM) --filepath FILEPATH
```

#### OPTIONS

##### Required:

```
--config    CONFIG_FILE e.g. coverity_pse_config.xml
--project    Use PROJECT to determine SCM plugin OR
--stream     Use STREAM to determine SCM plugin
--filepath   Lookup owner of given FILEPATH
```

##### Optional:

```
--host      CIM server HOST
--port      CIM server PORT
--username   CIM server USERNAME with admin access
--password   CIM server PASSWORD
--help      Print documentation and exit
```

#### EXAMPLE

```
perl test-scm-plugin.pl --config coverity_pse_config.xml --project
MYPROJECT --filepath /path/to/file.c
```

#### DESCRIPTION

This script can be used to test the loading and execution of an SCM plugin based on either a project or stream. It uses the plugin to perform an owner lookup based on the given filepath.

Once the plugin is verified as working, the assign-owners-to-unassigned-cids.pl script should work.

#### CONFIGURATION

This script uses a configuration file (usually coverity\_pse\_config.xml). There must be a system of type "scm" defined and the appropriate project or stream should be mapped to it. The system contains a <strip-path> tag that can be used to strip the correct prefix before the filepath is passed to the command execution.

#### USAGE RECOMMENDATION

This script is used to verify that the coverity\_pse\_config.xml is correct and the correct SCM plugin is loaded and functions as expected. It is provided as an aid to ensure things work before running assign-owner-to-unassigned-cids.pl.

The --filepath argument must be the exact filepath as it appears in CIM i.e. with forward slashes and all lowercase.



## 2) ASSIGN-OWNER-TO-UNASSIGNED-CIDS.PL

### SYNOPSIS

```
assign-owner-to-unassigned-cids.pl [options] --config CONFIG_FILE  
--project PROJECT (OR --stream STREAM1 --stream STREAM2 ...)
```

### OPTIONS

#### Required:

```
--config    CONFIG_FILE e.g. coverity_pse_config.xml  
--project   Assign ownership of defects in all streams belonging to a  
PROJECT OR  
--stream    Assign ownership of defects in STREAM(s)
```

#### Optional:

```
--host      CIM server HOST  
--port      CIM server PORT  
--username  CIM server USERNAME with admin access  
--password  CIM server PASSWORD  
--force     Set owner even if already set  
--dry-run   Test run, do not update defects  
--help      Print documentation and exit
```

### EXAMPLE

```
perl assign-owner-to-unassigned-cids.pl --config coverity_pse_config.xml  
--project MYPROJECT
```

### DESCRIPTION

This script assigns all CIDs of a project or stream(s) to owners based on the Software Configuration Management (SCM) system used.

Several plugins for different SCM systems (CVS, Subversion, ClearCase, and Accurev) are provided in the lib/Coverity/SCM directory. Additional plugins can be written and placed within this directory.

### CONFIGURATION

This script uses a configuration file (usually coverity\_pse\_config.xml). There must be a system of type "scm" defined and the appropriate project or stream(s) should be mapped to it.

### USAGE RECOMMENDATION

This script can be executed after each commit to the Integrity Manager database in order to assign defects to the last person who edited the affected file in version control.

### 3) SET-DEFECT-SEVERITY.PL

#### SYNOPSIS

```
set-defect-severity.pl [options] --mapping MAPPING_FILE
--project PROJECT (OR --stream STREAM1 --stream STREAM2 ...)
```

#### OPTIONS

##### Required:

```
--project    Set severity level in defects in all streams belonging to a
PROJECT OR
--stream     Set severity level in defects in STREAM(s)
```

##### Optional:

```
--config     CONFIG_FILE e.g. coverity_pse_config.xml OR
--host       CIM server HOST
--port       CIM server PORT
--username   CIM server USERNAME
--password   CIM server PASSWORD

--default-severity Default severity if not specified in the mapping file
--force       Apply severity even if already set
--dry-run    Test run, do not update defects
--help       Print documentation and exit
```

#### EXAMPLE

```
perl set-defect-severity.pl --config coverity_pse_config.xml --mapping
checker-severity-cxx.csv --project MYPROJECT
```

#### DESCRIPTION

This program will set the severity level of a project or stream's defects based on a checker to severity mapping file. By default, it will only set the severity of defects with an 'Unspecified' severity level.

#### CONFIGURATION

This script can use a configuration file (usually coverity\_pse\_config.xml) for CIM server information. The mapping file is CSV and must be modified to reflect the desired severity level for each checker prior to running the script.

#### USAGE RECOMMENDATION

This script can be executed after each commit to the Integrity Manager database in order to set the defect severity level.

#### 4) EMAIL-NOTIFY-PROJECT.PL

##### SYNOPSIS

```
perl email-notify-project.pl [options] --project PROJECT --to USERS
```

##### OPTIONS

###### Required:

```
--project  Report on new defects appearing in PROJECT
--to       CIM USERS, comma separated
```

###### Optional:

```
--config  CONFIG_FILE e.g. coverity_pse_config.xml OR
--host     CIM server HOST
--port     CIM server PORT
--username CIM server USERNAME with admin access
--password CIM server PASSWORD

--days    within number of DAYS, default 1 (24 hours)
--dry-run  Test run, do not send emails
--help     Print documentation and exit
```

##### EXAMPLE

```
perl email-notify-project.pl --config coverity_pse_config.xml --project
MYPROJECT --to admin,user1 --days 1 --dry-run
```

##### DESCRIPTION

Send an email to explicit CIM users reporting on New defects appearing in a project within the last N days. This script is not designed to operate explicitly on streams.

Emails are sent using the email facilities within CIM.

##### CONFIGURATION

This script can use a configuration file (usually coverity\_pse\_config.xml) for CIM server information.

##### USAGE RECOMMENDATION

This script can be executed after each commit to the Integrity Manager database so that particular users are alerted to all newly found defects.

## 5) EMAIL-NOTIFY-OWNERS.PL

### SYNOPSIS

```
perl email-notify-owners.pl [options] --project PROJECT
```

### OPTIONS

#### Required:

`--project` Report on defects assigned to owner in PROJECT

#### Optional:

`--config` CONFIG\_FILE e.g. coverity\_pse\_config.xml OR

`--host` CIM server HOST

`--port` CIM server PORT

`--username` CIM server USERNAME with admin access

`--password` CIM server PASSWORD

`--days` within number of DAYS, default 1 (24 hours)

`--dry-run` Test run, do not send emails

`--help` Print documentation and exit

### EXAMPLE

```
perl email-notify-owners.pl --config coverity_pse_config.xml --project  
MYPROJECT --days 1 --dry-run
```

### DESCRIPTION

Send an email to owners (i.e. CIM users) reporting on defects recently assigned to them within the last N days. This script is not designed to operate explicitly on streams.

Emails are sent using the email facilities within CIM.

### CONFIGURATION

This script can use a configuration file (usually coverity\_pse\_config.xml) for CIM server information.

### USAGE RECOMMENDATION

It is recommended that this script be run daily for each project so that users receive email alerts of defects assigned to them over the past 24 hours.

## 6) EXPORT-DEFECT-HANDLER.PL

### SYNOPSIS

```
export-defect-handler.pl [options] --config CONFIG_FILE XML_FILE
```

### OPTIONS

#### Required:

```
--config    CONFIG_FILE e.g. coverity_pse_config.xml
```

#### Optional:

```
--host      CIM server HOST  
--port      CIM server PORT  
--username  CIM server USERNAME  
--password  CIM server PASSWORD  
--force     Apply external reference even if already set  
--help      Print documentation and exit
```

### EXAMPLE

```
perl export-defect-handler.pl --config coverity_pse_config.xml  
path/to/cid.xml
```

### DESCRIPTION

This script will create a new issue in an external issue tracking system based on an xml file describing the defect auto-generated by CIM when the Export button is pressed. It then updates the CIM external reference field of the defect based on an identifier returned by the issue tracking system.

Several plugin for different issue tracking systems (Bugzilla, Jira, and ClearQuest) are provided in the lib/Coverity/IssueTracking directory. Additional plugins can be written and placed within this directory.

### CONFIGURATION

This script uses the coverity\_pse\_config.xml configuration file to obtain CIM connection information and information describing the issue tracking system.

### USAGE RECOMMENDATION

This script is installed in conjunction with an executable *export-defect-handler* (UNIX) or *export-defect-handler.bat* (Windows) file placed in the Integrity Manager bin directory.

## ABOUT COVERITY

---

Coverity, the leader in software integrity, is the trusted standard for companies that have a zero tolerance policy for software failures, problems, and security breaches. Coverity's award-winning portfolio of software integrity products, including the industry's leading static analysis solution, enables customers to Coverity software problems throughout the application lifecycle.

A privately held company headquartered in San Francisco, Coverity was founded in 2002 by leading Stanford University scientists whose four-year research project resulted in a breakthrough solution to one of the costliest problems in the software industry.

Today, Coverity has offices in 6 countries and employs more than 130 of the industry's brightest minds. Over 100,000 developers, 280 open source projects and 900 leading companies, including Juniper Networks, Research in Motion, Honeywell, Samsung, Symantec and Ericsson rely on Coverity to help them ensure the delivery of superior quality software and products.

