KDM Analytics

Working Together to Build Confidence

USER'S GUIDE

TOOL OUTPUT INTEGRATION FRAMEWORK

Document Version 1.6

Software Release 2.1.0

This document applies to Tool Output Integration Framework (TOIF) version 2.1.0 and may apply to subsequent releases. To check for newer versions of this document, please contact KDM Analytics Customer Support (e-mail: support@kdmanalytics.com).

While every attempt has been made to ensure that the information in this document is accurate and complete, some typographical errors or technical inaccuracies may exist. KDM Analytics does not accept responsibility for any kind of loss resulting from the use of information contained in this document.

Due to continued product development the information contained in this document may change without notice. Any improvements or changes to either the product or the document will be documented in subsequent versions.

KDM Analytics, Inc.
1956 Robertson Rd
Suite 204
Ottawa, ON
Canada
(613) 627-1010
Internet E-Mail: support@kdmanayltics.com
Website: http://www.kdmanalytics.com

Contents

Introduction

The *Tool Output Integration Framework* (TOIF) is a powerful open source vulnerability detection platform. It allows users to analyze systems utilizing multiple open source software (OSS) static code analysis (SCA) tools, and analyze the results in a common format using a single viewer. TOIF provides:

- Reference implementation for standard-based adaptors
- Standard-based normalization and reporting of vulnerabilities as Common Weakness Enumerations (CWEs), and their base Software Fault Patterns (SFPs)
- Utilization of open source development to advance the Software Assurance space
- A common protocol for exchanging vulnerability findings

TOIF is based on existing standard protocol for exchanging system facts, the OMG Knowledge Discovery Metamodel (KDM), now ISO/IEC 19506.

This document contains information about how to use TOIF. It includes procedures, notes, and other background information.

Audience

This document is intended to help system and software engineers, system analysts, security analysts, and system architects to use the *Tool Output Integration Framework* (TOIF) in performing defect sightings.

Typographical Conventions

Before you start using this guide, it is important to understand the terms and typographical conventions used in the documentation.

The table below identifies the formatting conventions used in KDM Analytics documents to represent different types of information.

Type of information	Formatting convention	Example
Slash characters in path names	Follow the UNIX convention (forward slash). Usually appears in `monospace font` as part of command-line input or output. **Note**: Substitute with a backward slash (\) for Windows platform.	`/workspace/OSS_TOIF`

File path and names	`monospace font`	Double click on the `eclipse.exe` file.
Information to be substituted with user-provided information	Description of information item to be supplied surrounded by angle brackets, `monospace font`	Extract the `eclipse-cpp-kepler-SR2-<platform version>`.zip file into a desired target folder.
Filename in descriptive text	*italics*	Locate the *com.kdmanalytics.toif.p2-2.1.0.zip* file
Step-by-step procedures.	1., 2., 3.,...	1) Click **START** 2) Click **Advanced** tab
User interface fields and menu options	**SMALL CAPITALS**	Right click on **MY COMPUTER** and from
User interface command buttons	**Bold**	1) Click **Start**
Names of keys on the keyboard.	CAPITALS	SHIFT, CTRL, or ALT.
Key combinations for which the user must press and hold down one key and then press another	KEY+KEY	CTRL+P, or ALT+F4.

For more information on specialized terms used in the documentation, see the Glossary at the end of this document.

Getting Help

A PDF version of the User Guide is available directly from the KDM Analytics product distribution zip file.

The following sections provide details on when and how to contact KDM Analytics support if you encounter a problem and cannot resolve the issue after consulting the published information.

When to contact us

If you encounter a problem or deficiency working with our product and are unable to resolve it after consulting the published information, please contact KDM Analytics support by e-mail: support@kdmanalytics.com.

Chapter 1

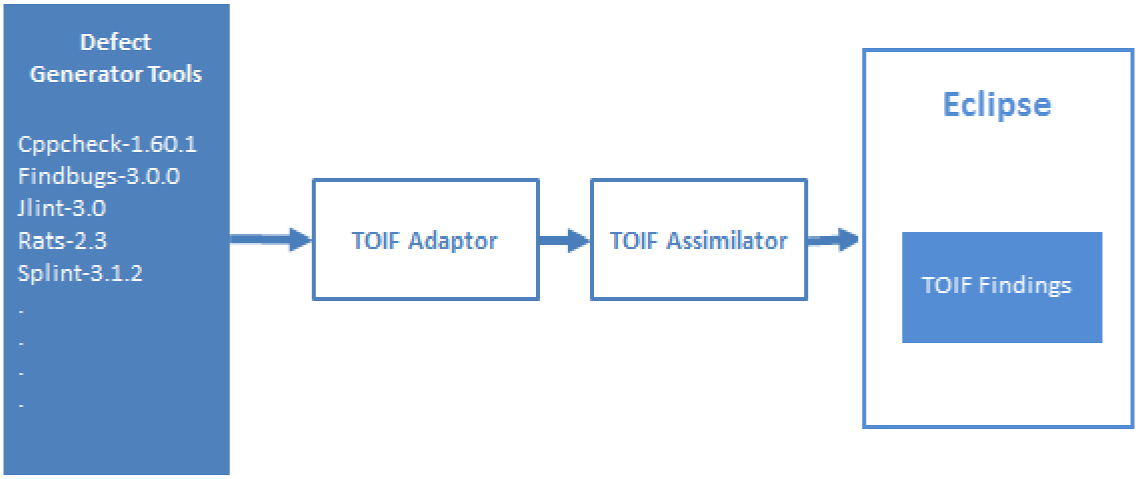
What is TOIF?

The *Tool Output Integration Framework* (TOIF) is a powerful open source vulnerability detection platform that provides analysts information of system defects with the ability to:

- Integrate multiple OSS SCA tools as “data feeds” into the repository
- Collate findings from several OSS SCA tools
- Put vulnerability findings into the context of other facts about the system (such as metrics, architecture, design patterns, etc.)

How does TOIF work?

TOIF takes the output of Open Source Software (OSS) Static Code Analysis (SCA) tools and displays the results in *Eclipse*.



TOIF Components

TOIF includes the following components:

- TOIF Adaptor: *TOIF Adaptor* is used to collect the output from various Open Source Software (OSS) Static Code Analysis (SCA) tools and convert their output into TOIF xml
- TOIF Assimilator: After running the *TOIF Adaptor* you need to run the Assimilator to merge TOIF findings and/or KDM data into a common fact-orientated repository or file
- TOIF Findings View: Once you have your TOIF findings assimilated you use the *TOIF Findings View* to display the results in *Eclipse*.

Chapter 2

Preparing to Install TOIF

The following sections provide the information required to install and get you working in the *Tool Output Integration Framework* (TOIF).

Installation Overview

This section provides the high-level steps for installing and running TOIF and viewing results in *Eclipse*.

1) Gather the installation packages for *TOIF RCP* and *TOIF Findings View*.

   The TOIF RCP package is called *kdmanalytics-oss-toif-2.1.0.linux.gtk.x86_64.tar.gz (linux)* and *kdmanalytics-oss-toif-2.1.0.win32.win32.x86_64.zip (windows)*. The *TOIF Findings* view is called *com.kdmanalytics.toif.p2-2.1.0.zip*.

2) Ensure that **Eclipse 4.3.2 (Kepler)*** has been installed (http://www.eclipse.org/downloads/packages/release/Kepler/SR2).
3) Read all of the information in this chapter before you install all the TOIF packages.
4) Unzip and install the *TOIF RCP* package.
5) Ensure that the supported OSS SCA tools you want to run TOIF with are installed. See **“Installing Open Source Software (OSS) Static Code Analysis (SCA) Tools”** on page 7 for more information.
6) Run the *TOIF Adaptor* with the desired Open Source Software (OSS) Static Code Analysis (SCA) tool.
7) Run the *TOIF Assimilator* against the TOIF files generated by the *TOIF Adaptor*.
8) Install the *TOIF Findings* view in your Eclipse instance and create a project.
9) Import the *.kdm file that was generated from running the TOIF Assimilator into an Eclipse project and view it in the *TOIF Findings* view in Eclipse.

*Note, the *TOIF Findings View* is only supported in Eclipse 4.3.2. Do not attempt to install the *TOIF Findings* view in another version of Eclipse.

Supported Deployment Configurations

The *TOIF RCP* is a standalone program. *TOIF Findings* view should be used within an instance of *Eclipse 4.3.2*.

System Requirements

To install TOIF your system must meet the minimum hardware and software requirements listed in the following sections.

Client Hardware

The following table lists the supported hardware platforms for TOIF.

Minimum	Recommended
500 MB free hard drive space	
2 GB RAM	8 GB RAM
Dual Core Processor	Quad core processor

Client Software

The following table lists the supported operating system platforms for TOIF.

Platform	Recommended
Ubuntu 14.04.1 (64 bit)	Ubuntu 14.04.1 (64 bit)
Fedora 17.x (64 bit)	
Microsoft Windows 7 (64 bit)	Microsoft Windows 7 (64 bit)
Eclipse 4.3.2 (Kepler)	Eclipse 4.3.2 (Kepler)

Chapter 3

Installing TOIF Packages

To install the *TOIF RCP* for linux or windows and the *TOIF Findings View* packages perform the following steps.

1) Download the `kdmanalytics-oss-toif-2.1.0.linux.gtk.x86_64.tar.gz (linux)` or `kdmanalytics-oss-toif-2.1.0.win32.win32.x86_64.zip (windows)`, and `com.kdmanalytics.toif.p2-2.1.0.zip` files.

   Note: These files are located on the release page of GitHub and the KDM Analytics, Inc. website (www.kdmanalytics.com/toif/download.html).

2) Run an unzip utility to extract the `kdmanalytics-oss-toif-2.1.0.linux.gtk.x86_64.tar.gz (linux)` or `kdmanalytics-oss-toif-2.1.0.win32.win32.x86_64.zip (windows)` into a desired target folder.

Installing Open Source Software (OSS) Static Code Analysis (SCA) Tools

Install the following open source software (OSS) static code analysis (SCA) tools according to their own instructions. The supported input file types are listed beside each OSS SCA tool.

- **Cppcheck-1.60.1:** .c and .cpp files
- **Findbugs-3.0.0:** .class files
  - **Find Security Bugs Plugin-1.2.1:** .class files

  Note: The Find Security Bugs Plugin is installed by placing the *findsecbugs-plugin-1.2.1.jar* in the Findbugs 3.0.0 plugins folder.
- **Jlint-3.0 (Ubuntu and Windows only):** .class files
- **Rats-2.3:** .c and .cpp files
- **Splint-3.1.2:** .c files

Make sure that you are using the release 2.1.0 of the TOIF RCP application with these versions of the OSS SCA tools. The executable for each tool should be on the system path. A simple check would be to open a command prompt and type the following for each respective OSS SCA tool: *cppcheck --version, splint --version*, *findbugs --version*, *jlint –help*, and *rats*. The system will respond after each command with either a version number, for example, *Cppcheck 1.60.1*, or in the case of *jlint -help* some information that describes available options and message categories and in the case of rats some run information such as entries and Total lines analyzed.

Chapter 4

Running the TOIF Adaptor

This section will discuss the possible ways of running the TOIF Adaptor:

- Running the TOIF Adaptor from command line
- Integrating with a C or Java project's build

Running the TOIF Adaptor from the command line

To run the Adaptor from a command line, perform the following steps.

1. Open a command prompt.
2. Make sure that the windows or linux TOIF RCP (toif) command is on the PATH. Also, ensure that the OSS SCA tools are correctly installed.
3. Type `toif --adaptor=<Adaptor Name> --inputfile=<full path to input file> --outputdirectory=<path to output directory> --housekeeping=<path to housekeeping file> -- [Additional arguments]`

Where:

- `<Adaptor Name>` defines the name of the adaptor class. This is the adaptor that is to be used with the input source file. From this class, the framework is able to discover housekeeping facts about the adaptor as well as which OSS SCA tool to call and what options to use.
- `<full path to input file>` defines the full path to the input source file. In order for the adaptors to create all the facts for this file, a full path must be provided.
- `<path to output directory>` defines the path to the output directory. This is the directory where the TOIF XML file will be written.
- `<path to housekeeping file>` defines the path to the file containing the facts about the project's housekeeping. This file is specific to each adaptor and each project. This is because it is down to the user to provide the project details as well as which OSS SCA tool is running on the system. An example Housekeeping file is located in the *Examples* directory.
- `[Additional Arguments]` defines any additional arguments that you may want. These must be entered after the TOIF Adaptor's required arguments and after a “--”. These arguments can be included files or compilation options, and they will vary from tool to tool. For example, splint can take -I and -D options:

```
./toif --adaptor=splint --inputfile=/home/user/foo.c --
outputdirectory=/home/user/toifFiles --
housekeeping=housekeepingFile.txt -- -I./includes -D_U_=
```

Warning: Do not copy the commands from the manual as the “--“ may be pasted as “—“ in the command prompt window and will cause an “unexpected argument” error.

Integrating with C project's build

The best way to integrate the adaptors into the build is by wrapping the compiler and the adaptors into a script. When the compiler is called, the adaptors will be run for every source file used. To get the build process to use this wrapper instead of the compiler on its own, the compiler flag needs to be set during the make:

```
./configure
```

The make can then be continued with configuration as:

- `make CC=<path to myGccWrapper>`
- `make install`

An example script has been provided in the Examples folder. However, the following directories will need to be changed within the script to suit your system.

- HOUSE_KEEPING = `<the location of the housekeeping file>`
- OUTPUT_DIR = `<the output directory for the toif files>`

Integrating with Java project's build

It may be possible to integrate into a Java project's build by adding the following to the *build.xml* file.

```
<!-- my target -->
  <target name="mytarget" depends="build">
    <apply executable="python">
      <fileset dir="${build.dir}">
        <patternset>
          <include name="**/*.class"/>
        </patternset>
      </fileset>
      <arg value="/TOIF/javaAdaptors.py"/>
      <srcfile/>
    </apply>
  </target>
```

This creates a new target which will find all the *.class* files in the destination directory of the project. For each file, the *javaAdaptors* python script will be run with the arguments that are specified. An example script is provided in the Examples folder in the TOIF RCP installation directory. The script is for reference only and you will have to write your own to be compatible with your system and project.

Chapter 5

Running the TOIF Assimilator

The *TOIF Assimilator* merges TOIF findings (toif files created by running the TOIF Adaptor) and/or KDM data into a common fact-orientated repository or file.

To run the *TOIF Assimilator*, perform the following steps.

1. Open a command prompt.
2. Make sure that the windows or linux TOIF RCP (toif) is on the PATH.

   Do the following:

   - Type `toif --merge --kdmfile=<output destination> --inputfile=[files or directories to merge...]`.

   Note: The file extension for the output destination must be `.kdm`.

   Where:

   - `<output destination>` The path and filename of the .kdm file should be specified as the destination.
   - `[files or directories to merge...]` defines the toif files to be merged. Any number of toif files can be entered here. Alternatively, a directory can be specified which contains the toif files.

   For example,

   ```
   ./toif --merge --kdmfile=/home/user/outputFile.kdm --
   inputfile=/home/user/toifFiles/
   ```

   Warnings:

   Ensure that the output file is not produced where the input files are being read from.

   If running the assimilator on large projects with many files, the default toif memory setting may need to be increased. Memory allocation can be updated in the *toif.ini* file, located in the toif install directory, by modifying the "-Xmx" parameter.

   We recommend that you do not copy the commands from the manual as the "--" may be pasted as "—" in the command prompt window and will cause "unexpected argument" errors.

The resulting assimilated output data, from running the *TOIF Assimilator*, is used as the input to the *TOIF Findings View*.

Note: The output of the TOIF Assimilator, *<output file name>.kdm,* is a zip file (the .zip extension may not be visible) and you will need to extract it to a <output file name> directory and use the *.kdm file that resides in the directory to complete the subsequent steps. For example, if your output file is *outputFile.kdm.zip* you would extract it to a directory named *outputFile* and in this directory you would have an *outputFile.kdm* file.

Chapter 6

TOIF Findings View

The *TOIF Findings* view allows the assimilated output data to be viewed in *Eclipse*. The output data can be used by analysts who are performing defect sightings on a project.

Installing the TOIF Findings View in Eclipse

To install the *TOIF Findings* view in *Eclipse 4.3.2* perform the following steps.

1) Start Eclipse.

2) Choose the workspace location.

   This is home to the eclipse user/session data and also any projects which are created.

3) Close the welcome screen.

4) If you have a previous version of the TOIF Findings View installed, you will need to uninstall it prior to installing a new one.

   Note: Please ensure that in addition to the uninstallation, you have also removed the old release from the “Available Software Sites” list.

5) Click HELP in the tool bar menu and from the drop down menu select INSTALL NEW SOFTWARE....

   The *Install* dialog opens.

6) Click **Add…**.

   The *Add Repository* dialog opens.

7) Click in the NAME: text field and enter a name for the new software source.

   For example, *TOIF OSS*.

8) Click **Archive…**.

   The *Repository Archive* dialog opens.

9) Navigate the directories to find the location of the `com.kdmanalytics.toif.p2-2.1.0.zip` file and then click **Open**.

   The *Repository Archive* dialog closes and the *Location:* text field in the *Add Repository* dialog is populated.

10) Click **OK** in the **Add Repository** dialog.

    The *Add Repository* dialog closes and the *SFP/CWE* category appears under the *Name* column in the *Available Software* panel.

Note: The category is displayed only if the *Group items in category* check box is selected. Otherwise, the single feature, *TOIF* is displayed.

11) Click SFP/CWE to expand the category in the **Available Software** panel.

The feature, *TOIF* is displayed.

12) Click to select the *TOIF* feature.

13) Click **Next >**.

The *Install Details* panel is displayed.

14) Click **Finish**.

The *Install* dialog closes and the *TOIF Findings View* installation starts. Once the installation is complete a dialog will appear requesting that Eclipse be re-started.

Note: If a dialog appears with a warning that the content is unsigned simply click *OK*. This allows the *TOIF Findings View* installation to continue.

15) Click **Yes**.

Eclipse session will close and then re-open.

16) Click HELP in the Eclipse menu bar and from the drop down menu select ABOUT ECLIPSE.

The *About Eclipse* dialog opens.

17) Click the **Installation Details** button.

The Eclipse Installation Details dialog opens.

18) Click the **Installed Software** tab in the **Eclipse Installation Details** dialog and ensure that the TOIF feature is listed.

If the *TOIF* feature exists then the *TOIF Findings View* has been installed into Eclipse.

Importing Data into the TOIF Findings View in Eclipse

To import the assimilated output data into the *TOIF Findings View* in *Eclipse* perform the following steps.

1) Click FILE -> NEW -> PROJECT... in the Eclipse menu bar.

The *New Project* dialog opens allowing you to create a new project.

2) Click **General** folder to expand it and then select **Project** from the list

3) Click **Next>** in the **New Project** dialog.

The *New Project* dialog updates to display the Project panel.

4) Click in the PROJECT NAME: text field and type the name you want to give to the project.

For example, *TOIF Project*.

5) Click **Finish** at the bottom of the **New Project** dialog.

The project is displayed in the Navigator panel of the Eclipse session.

6) Right click on the project in the Navigator panel and from the drop down menu click IMPORT....

The *Import* dialog opens.

7) Expand the SFP/CWE folder and navigate to IMPORT INTEGRATED SFP/CWE FILE.

8) Click IMPORT INTEGRATED SFP/CWE FILE.

The *Import Integrated SFP/CWE File* is now selected.

9) Click **Next>** in the **Import** dialog.

The *Import* dialog updates.

10) Select the project that you just created in the list currently displayed in the Import dialog.

11) Click the **Browse…**.

A dialog opens.

12) Navigate the directory to find the TOIF Data and select the *.kdm* file, the output of running the **TOIF Assimilator** (see page 11).

13) Click **Open**.

The dialog closes.

14) Click **Finish**.

The *Import* dialog closes and the *Import SFP/CWE Data* dialog appears to display the progress of importing the SFP/CWE data.

15) Click WINDOW -> SHOW VIEW -> OTHER... in the Eclipse menu bar.

The *Show View* dialog opens

16) Click SFP/CWE folder and navigate to find TOIF FINDINGS.

17) Click TOIF FINDINGS to select it.

18) Click **OK**.

The *Show View* dialog closes and after the model is finished loading, the *TOIF Findings View* appears in the *Eclipse* List panel. The defect data is populating the *TOIF Findings View*.

If you want to view the source code of the file associated with a finding you need to first add the project sources to the applicable TOIF project in Eclipse. After adding the project sources you can double click on the finding in the *TOIF Report View* and the source code will be displayed in your default source code viewer.

Scoping Defect Findings in the TOIF Findings View

Defect findings currently displayed in the *TOIF Findings* View can be scoped to better to focus your analysis and citing by applying a term search and/or a filter option. Scoping can be applied in the following ways:

- Search and term filter
- Filter options
- Sorting by column headings
- Filter on a selected project source file(s)
- Any combination of the above

Scoping can be performed before or after ***you analyze and cite*** (see "***Analyzing and Citing Defect Findings in the Findings View***" on page 18) your defect findings.

When any of the scoping methods are applied the TOIF Findings view will indicate that a filter(s) is applied by displaying a "**(Filter(s) active)**" message in red in the label above the defect findings table. For example, *[MyProject]* **(Filter(s) active)** *Number of defects: 2 (500 filtered from view)*.

Search and term filter

To do this, perform the following steps:

1) Click on the project that contains defect findings in the Project Explorer.

   The project is highlighted to indicate that it is selected and the respective defect findings are displayed in the *TOIF Findings* view.

2) Click in the Search and term text field in the *TOIF* Findings view and type a term.

3) Click **Search**.

   Only findings that contain the specific term(s) will be displayed in the *TOIF* Findings View. Running the search without any terms in the term filter field will show all the findings in the data set.

   Note: To clear the Search and term text field click the *Clear* button in the *TOIF* Findings view.

Filter Options

To do this, perform the following steps:

1) Click on the project that contains defect findings in the Project Explorer.

   The project is highlighted to indicate that it is selected and the respective defect findings are displayed in the *TOIF Findings* view.

2) Click Filter ▽ icon in the *TOIF Findings* view toolbar and from the drop down menu select FilterS….

The Filter window is displayed.

3) Do any combination of the following:

a) Click the check box in front of 2+ Tools report same location to view all defect findings reported on the same file and line number by two or more OSS SCA tools.

b) Click the check box in front of 2+ Tools report same location. With the same CWE. to view all defect findings reported on the same file and line number with the same CWE by two or more OSS SCA tools.

c) Click the check box in front of 2+ Tools report same location. With the same SFP. to view all defect findings reported on the same file and line number with the same SFP by two or more OSS SCA tools.

d) Click the check box in front of Trust above: and enter a value between 0 and 100 in the text field to view all defect findings with a trust level above the value entered in the test field.

e) Click the check box in front of Is Valid to view all defect findings cited as Is a Weakness or "true" citing.

f) Click the check box in front of Not Valid to view all defect findings cited as Not a Weakness or "false" citing, or are unmarked.

g) Click the check box in front of Not SFP--1 to view all security related defect findings.

Note: The filtering is applied to the defect findings currently displayed in the *TOIF Findings* view.

4) Click **Ok**.

The *Filter* window closes and the *TOIF Findings* view updates to display only the defect findings associated with the applied filtering options.

Note: If the *Filter* option is applied in combination with the *Search and term* filter the *Filter* option(s) is applied to the defect findings currently displayed in the *TOIF Findings* view.

Sorting by Column Heading

To do this, perform the following steps:

1) Click on the project that contains defect findings in the Project Explorer.

The project is highlighted to indicate that it is selected and the respective defect findings are displayed in the *TOIF Findings* view.

2) Click on any of the heading columns to sort the defect findings alpha-numerically by column.

Filter on a Selected Project Source File(s)

To do this, perform the following steps:

1) Navigate to a project source file(s) that contains defect findings in the Project Explorer.

   The project source file(s) is highlighted to indicate that it is selected and the respective defect findings are displayed in the *TOIF Findings* view.

   Note: Hold **CTRL** key and click the project source files to choose them.

2) Right click on the project source file and select Filter TOIF Findings on Selection in the drop down menu.

   The *TOIF Findings* view updates to display defect findings associated with the selected project source file(s).

Analyzing and Citing Defect Findings in the TOIF Findings View

The source code associated with the defect finding can be analyzed and then cited in the *TOIF Findings* view. To do this do the following.

1) Click on a project that contains defect findings in the Project Explorer.

   The project is highlighted to indicate that it is selected and the respective defect findings are displayed in the *TOIF Findings* view.

2) (Optional) ***Scoping Defect Findings in the Findings View*** on page 18 to narrow the focus of your analysis.

3) (Optional) Select a defect finding instance in the **TOIF Findings** view and click the **Defect Description** icon in the **TOIF Findings** view toolbar.

   The *Defect Description* tab opens to provide the cluster, SFP, and CWE description of the selected defect instance.

4) Double click on a defect finding instance in the *TOIF Findings* view.

   The selected file opens to the location of the defect in the Editor panel of Eclipse. Place the cursor over the marker displayed to the left of the finding to pop up a tool tip with the defect description.

   Note: If a *"Cannot open file"* dialog appears it is because the source file associated with the defect finding does not exist in your project. Please check to ensure that the file exists in your project and has not been deleted.

5) Analyze the defect and do any of the following to cite the defect:

   a) If you decide that the defect is a weakness right click anywhere on the finding instance in the *TOIF Findings* view and select Is a Weakness from the drop down menu.

A red “**X**” appears beside the Tool, SFP, and CWE column cells within the *TOIF Findings* view for the respective finding instance.

b) If you decide that the defect is not a weakness right click anywhere on the finding instance in the *TOIF Findings* view and select Not a Weakness from the drop down menu.

A green checkmark appears beside the Tool, SFP, and CWE column cells within *the TOIF F*indings view for the respective finding instance.

c) If you decide that the OSS SCA tool's ability to accurately detect the defect finding is suspect you can set the trust level value between 0-100 to indicate the level of confidence. This level is propagated throughout the data set, marking any finding with the same CWE from the same tool with the specified trust level value.

Note 1: When a project is cited for the first time a warning dialog appears with the following message:

Warning: Citings are file attributes. Editing or deleting a file will delete its citing information. Daily snapshots of citing information are saved in <project>/.KDM/TOIF/historystating

Click **Ok** in the Warning dialog. The Warning dialog closes and the initial citing is applied.

Note 2: If you need more information on the defect and you have a connection to the internet right click on the defect finding instance in the **TOIF Findings** view and select More Information from the drop down menu. A browser window opens displaying the mitre site for the selected CWE.

Exporting cited defect findings to a *.tsv file

Defect findings, including citing information, can be exported to a *.tsv file. To do this, perform the following steps.

1) Click on the project that contains defect findings in the Project Explorer.

The project is highlighted to indicate that it is selected and the respective defect findings are displayed in the *TOIF Findings* view.

2) Click on any defect finding instance or press **CTRL+A** to select all instances in the *TOIF Findings* view.

All defect findings in the *TOIF Findings* view are selected.

3) Click **Export Selection** icon in the *TOIF Findings* view tool bar menu.

A *Save As* dialog opens.

4) Click in the File name: text field and enter the name of the file.

5) Click **Save**.

The *Save As* dialog closes and a *<filename>.tsv* file is saved to the specified directory.

Note: TOIF automatically saves a <project name>-.<yyy-mm-dd>T<hh-mm-sec>.tsv file every time citing information is modified. The file is located in <project workspace directory>/<project name>/.KDM/TOIF/history directory.

Importing cited defect findings from a *.tsv file

Defect findings, including citing information, exported from one TOIF project can be imported to another TOIF project by a *.tsv file. To do this, perform the following steps.

Note: The projects that you are importing the files to must have the cited source files for the citings to be applied.

1) Click on the project that contains the same source files as the project the ***.tsv file** (see "***Exporting cited defect findings to a *.tsv file***" on page 19) was exported from.

   The project appears in the *Project Explorer* and the defect findings are displayed in the *Findings* view.

2) Right-click on the project and select IMPORT... from the drop-down menu.

   An *Import* dialog opens.

3) Expand the **SFP/CWE** folder and select **Import Citing File (*.tsv)**.

   The *Next>* button becomes enabled.

4) Click **Next>**.

   The *File Import Wizard* appears.

5) Select the project from the *Select target project:* list.

6) Click the **Browse** button and navigate to and select the **.tsv* file.

   The location to the *.tsv file populates the *Select TOIF Data:* text field and the *Finish* button becomes enabled.

7) Click **Finish**.

   Note: If your project does not contain the same source files related to the *.tsv file then no finding or citing information will be displayed in the *TOIF Findings* view.

8) Select the *TOIF* **Findings** view.

   The defect findings and any citing information from the imported *.tsv file is displayed in the *TOIF Findings* view.

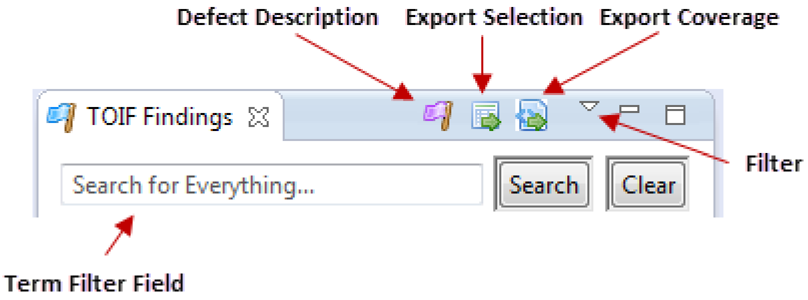
Chapter 7

TOIF Findings View Interface

The following section provides the descriptions of the toolbar buttons and context-sensitive menu options in the *TOIF Report View* that can be used in performing a defect analysis on your project.

TOIF Findings View toolbar

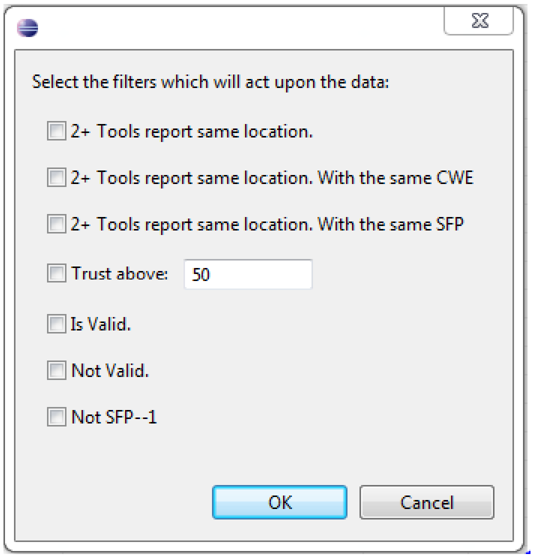
The *TOIF Findings* View consists of a toolbar which is located in the top right corner contains the following buttons and fields:



Filter

Filters reduce the number of visible findings on the screen.

- The 2+ Tools filter options only displays findings where two tools found a finding in the same location, same location with the same CWE, or same location with the same SFP.
- The trust filter only shows findings with a trust above the set amount.
- The "Is valid" filter option displays the findings that have been marked as Is a Weakness
- The "Not valid" filter option displays the findings that have been marked as either Not a Weakness or are unmarked.
- The Not SFP--1 filter option removes all non-security related (SFP--1) defect findings from the TOIF Findings view.



Select the filters which will act upon the data:

□ 2+ Tools report same location.

□ 2+ Tools report same location. With the same CWE

□ 2+ Tools report same location. With the same SFP

□ Trust above: 50

□ Is Valid.

□ Not Valid.

□ Not SFP--1

Clicking in the check box in front of a filter option toggles a check mark; displayed or not displayed. A check mark in the check box denotes that the filter option is selected. No check mark denotes that filter option is not selected. Only selected filter options will be applied.

Export Selection

Exports the selected elements to a *.tsv format. This file can then be imported into an Eclipse project containing the same source files as the project used to export the *.tsv or programs such as Excel or Libre Office Calc. It is important to note that this file is tab separated.

Export Coverage

Exports the entire data set as a Coverage Claims Representation (CCR). More information for this coverage report can be found at http://cwe.mitre.org/compatible/ccr.html.

Defect Description

Displays the cluster, SFP, and CWE description of the selected defect instance.

Search and Term Filter Field

The *term filter field* is a search box for findings containing a specific term. Clicking the **Search** button executes the search on the terms included in CWE/SFP ids, Description contents, line numbers, OSS SCA Tool name, or resource names of the findings. Only findings that contain the specific term(s) will be displayed in the TOIF Findings View. Running the search without any terms in the term filter field will show all the findings in the data set.

Clicking the **Clear** button removes the terms from the *term filter field* and updates the *TOIF Findings* view to display the list of defect findings displayed prior to executing the search.

TOIF Findings View Context Sensitive Menu Options

Right clicking on data displayed in the *TOIF Findings View* will display a drop down menu with the following options.

Not a Weakness

This marks that the finding is not actually a weakness.

Is a Weakness

This marks that the finding is a weakness.

Uncite Weakness

This unmarks a finding that was previously marked as either "Not a Weakness" or "Is A Weakness".

Set Trust Level

This option is only available at the "Finding" level. It sets the level of trust for the selected finding. This level is propagated throughout the data set, marking any finding with the same CWE from the same tool with the specified value. Trust is an indication of how much faith the analyst has in the tools ability to accurately detect the defect.

Trace

This option is only available at the "Finding" level. If trace data is present, selecting this option will display the trace back as a dynamic menu which when clicked will take you to the various places within the code; tracing the route all the way to where the finding was generated.

More Information

This will take you to the mitre site for the selected CWE id.

TOIF Findings View Sorting

Click on the column headings to alphabetically sort in ascending and descending order

Chapter 8

Known Limitations

Traceback displays numbers as reported by TOIF Adaptors

The traceback currently shows numbers as reported by the TOIF Adaptors unlike the code locations in the view list (which are normalized to kdm locations).

Jlint not supported on Fedora platform

Use of Jlint 3.0 OSS SCA tool with TOIF 2.1.0 is not supported on the Fedora-17.x 64-bit operating system.

Glossary of Terms

C

Common Weakness Enumeration

Common Weakness Enumeration (CWE) is a software community project whose goal is to create a catalog of software weaknesses and vulnerabilities.

Coverage Claims Representation

Coverage Claims Representation (CCR) is an XML document used for representing information about Common Weakness Enumeration.

O

OSS SCA

An acronym for open source software static code analysis (OSS SCA).

S

Software Fault Patterns

Software Fault Patterns (SFP) are a generalized description of an identifiable family of computations:

- Described as patterns with an invariant core and variant parts
- Aligned with injury
- Aligned with operational views and risk through events
- Fully identifiable in code (discernible)
- Aligned with CWE
- With formally defined characteristics

Index