

# SIP Modification Checker

## Technical Reference

Version 1.00.00

Authors	Markus Schwarz
Status	Released

## Document Information

### History

Author	Date	Version	Remarks
Markus Schwarz	2014-05-07	1.00.00	Initial version

### Reference Documents

No.	Source	Title	Version
[1]			

### Scope of the Document

This technical reference describes the general use of the tool SipModificationChecker.

## Contents

<b>1</b>	<b>Component History .....</b>	<b>5</b>
<b>2</b>	<b>Introduction.....</b>	<b>6</b>
2.1	Overview.....	6
2.2	Background.....	6
2.3	Workflow .....	7
2.3.1	At Vector .....	7
2.3.2	At Tier1/OEM .....	7
<b>3</b>	<b>Functional Description .....</b>	<b>8</b>
3.1	Command Line Usage .....	8
3.1.1	Console Output.....	8
3.1.2	Return Error Codes.....	8
3.1.3	Report Creation.....	9
3.2	Integration.....	9
<b>4</b>	<b>Glossary and Abbreviations .....</b>	<b>10</b>
4.1	Glossary .....	10
4.2	Abbreviations .....	10
<b>5</b>	<b>Contact.....</b>	<b>11</b>

**Illustrations**

Figure 2-1	Workflow at Vector.....	7
Figure 2-2	Workflow at Tier1/OEM.....	7

**Tables**

Table 1-1	Component history.....	5
Table 3-1	Command Line ErrorCodes .....	8
Table 7-1	Glossary .....	10
Table 7-2	Abbreviations.....	10

## 1 Component History

The component history gives an overview over the important milestones that are supported in the different versions of the component.

Component Version	New Features
1.00.00	Initial Version

Table 1-1 Component history

## 2 Introduction

This document describes the functionality of the tool SipModificationChecker.

### 2.1 Overview

The SipModificationChecker is a tool that supports the Tier1 and OEM to determine if and where the (source code) files delivered from Vector were unintentionally changed.

The SipModificationChecker

- > is a command line based tool.
- > checks sources below a user-selectable root directory against information given in a reference file.
- > checks if and which of the relevant delivered files are contained below that directory.
- > checks if and which files have been modified.
- > reports the “modification state” as ERRORLEVEL and within a HTML report.

### 2.2 Background

This tool aims to find changes that were unintendedly introduced by the customer in the delivered Vector source code.

## 2.3 Workflow

### 2.3.1 At Vector

- > Vector creates the list of relevant files, i.e. all (source code) files that are relevant to not be modified
- > Vector calculates a check code for each of these files and creates the SipCheckCodeFile
- > Vector delivers the embedded sources, the SipCheckCodeFile and the SipModificationChecker

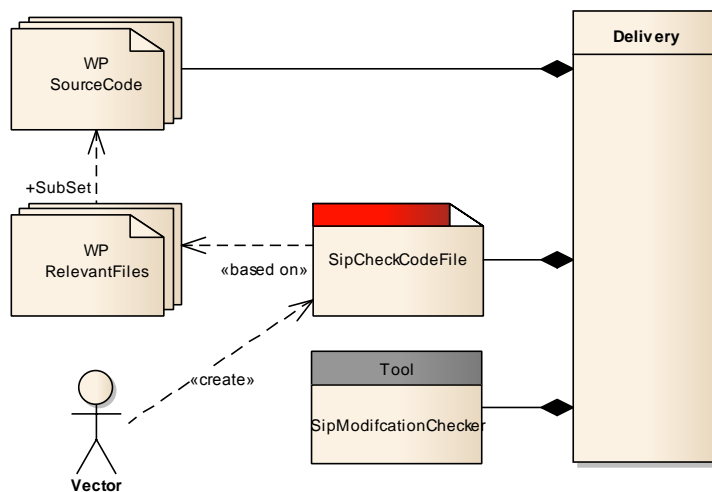


Figure 2-1 Workflow at Vector

### 2.3.2 At Tier1/OEM

- > The Tier1/OEM run the SipModificationChecker, configure their root path and load the SipCheckCodeFile from Vector. The tool provides information if the local sources found below the selected root directory are un-modified.

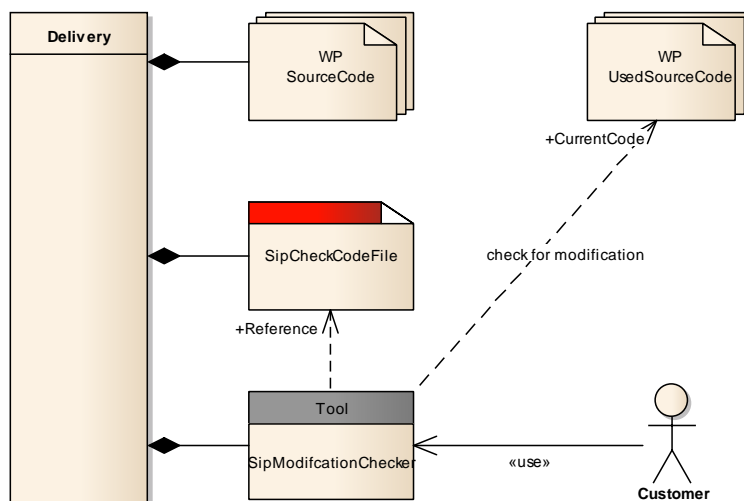


Figure 2-2 Workflow at Tier1/OEM

## 3 Functional Description

### 3.1 Command Line Usage

The analysis can be executed from command line.

Syntax:

```
SipModificationChecker <rootPath> <referenceFile>
  rootPath:      path to the directory that is used as base for analysis
  referenceFile: path to the provided reference file
```

Example:

```
SipModificationChecker c:\Example\CodeRootPath
C:\Example\SipModificationChecker.xml
```

If the rootPath or the referenceFile do not exist, the tool outputs a command line message and returns a ProcessError.

Otherwise the tool checks all files provided in referenceFile if they can be found below the rootPath and if they have local modifications compared to the delivery.

#### 3.1.1 Console Output

The tool outputs the results in the console:

```
=> Result:
    115/310 files are OK (unmodified)
    185/310 files are NOT OK (i.e. they have been modified)
    10/310 files are not found in given directory
    Refer to HTML for details
=> ERROR
```

#### 3.1.2 Return Error Codes

The tool sets the environmental variable ERRORLEVEL depending on its result:

Return Code	Value	Description
Ok	0	All referenced files are found. They have no modifications.
Warning	1	Not all referenced files are found. All found files have no modifications.
Error	10	There is at least one file with modifications.
ProcessError	20	The check could not be performed as the input data is not valid (rootPath and/or referenceFile do not exist)

Table 3-1 Command Line Error Codes



### 3.1.3 Report Creation

In addition to the returned error code, the tool creates a HTML report that indicates the status of each file.

Example:

#### Report on modified files

**2014-05-07 09:28:25**

This report aims to give you hints if and where you have unintended modifications of code files delivered by Vector. It lists all relevant source code files delivered by Vector. For each file it is reported if and where the file is found in your project directory and if the file is unmodified.

Values of Status:

- ok: The file is used without modifications.
- warn: The file is not used. (or at least not found)
- error: The file is used but has modifications.

#### Result

Vector has referenced 310 files on 2014-04-14 11:44:13.  
 You are using 115 files without modifications.  
 You are using 185 files with modifications.  
 You are not using 10 files.

#### File Overview

FileName	Location	Status
Adc.c	<a href="#">C:\Example\CodeRootPath\Adc.c</a>	ok
Adc.h	<a href="#">C:\Example\CodeRootPath\Adc.h</a>	ok
Bsd_Types.h	Original file: .\BSW\Bsd_Types.h	not used
BswM.c	<a href="#">C:\Example\CodeRootPath\BswM.c</a>	modified
BswM.h	<a href="#">C:\Example\CodeRootPath\BswM.h</a>	ok

The HTML report is located in the same directory as referenceFile. It is named like the referenceFile extended with “\_Report.html”.

Example:

The Reference File C:\Example\SipModificationChecker.xml  
 leads to report C:\Example\SipModificationChecker\_Report.html

### 3.2 Integration

It is recommended to integrate this tooling into the build environment to get early notifications of modified sources. This also prevents that the different sources are used for build and check.

Use the returned ERRORLEVEL to react on modified files, e.g. abort the build process or inform the user within the build report.

## 4 Glossary and Abbreviations

### 4.1 Glossary

Term	Description

Table 4-1 Glossary

### 4.2 Abbreviations

Abbreviation	Description
SIP	Software Integration Package

Table 4-2 Abbreviations

## 5 Contact

Visit our website for more information on

- > News
- > Products
- > Demo software
- > Support
- > Training data
- > Addresses

[www.vector.com](http://www.vector.com)