

Importing/exporting requirements

1 Abstract

This document provides information about import/export formats of requirement documents and the handling of the import process.

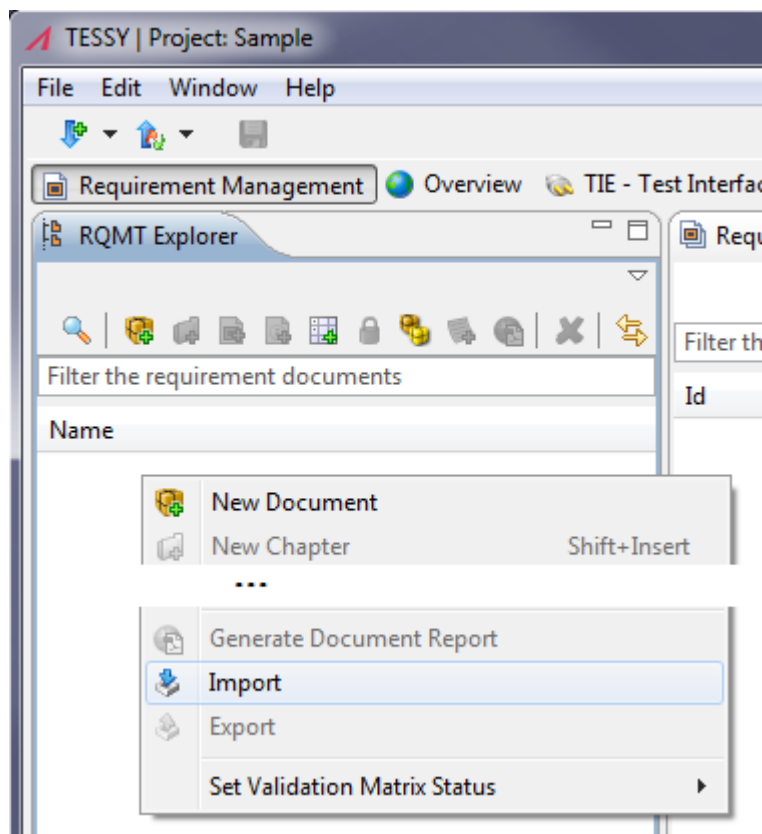
Table of contents

1	Abstract	1
2	Introduction.....	2
3	Import dialog settings	2
4	Import formats	3
4.1	XML format extensions.....	4
4.2	ID handling	4
4.3	Version handling.....	4
4.4	ReqIF format	4
4.5	Examples	5
4.5.1	CSV/TSV.....	5
4.5.2	XML.....	5
4.6	XML DTD format description	6
5	Exporting ReqIF documents	6
5.1	Creating additional result attributes	7
5.2	Updating the result attributes	7

2 Introduction

TESSY provides import and export capabilities for requirement documents. Requirements can be imported as a flat list (using the TXT and CSV format) and as structured documents containing chapters and requirements (using the XML or the ReqIF format).

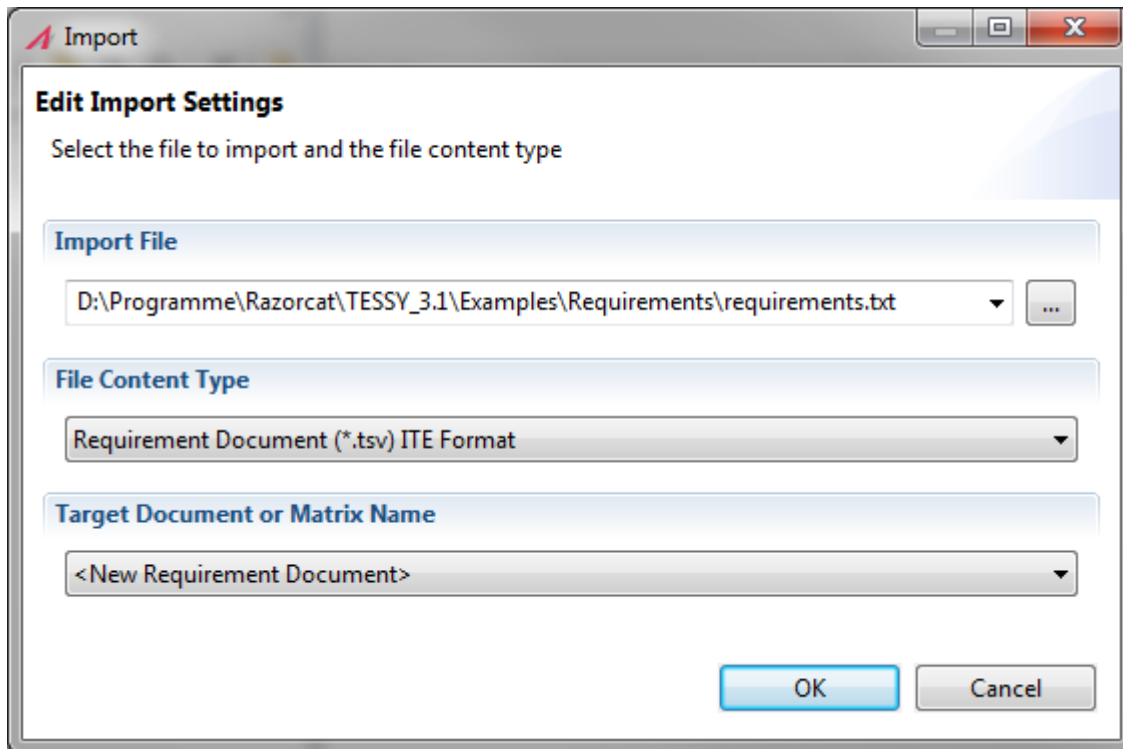
The import/export of requirement documents is available via the context menu of the requirement explorer within the requirement management perspective.



You can import requirement documents and verification matrices.

3 Import dialog settings

Within the import dialog you choose the import file and TESSY will automatically detect the corresponding import format. If there are errors within the import file or the format cannot be recognized, a warning message will be displayed.



You can select an existing document as target document or let TESSY create a new requirement document with the contents of the import file.

4 Import formats

The import dialog accepts files with the following CSV/TSV or XML format:

Column Name (CSV/TSV)	XML Attribute	Description
RequirementID	id	Id of the requirement (string, optional)
RequirementVersion	version	Version of the requirement (string of the form "1.0", optional)
RequirementShortDescription	shortdescription	A short description (or title) of the requirement (string, optional)
RequirementText	text	The text of the requirement (string)
RequirementPriority	priority	Priority (one of VERY_LOW, LOW, MEDIUM, HIGH, VERY_HIGH, optional)
RequirementDocument	-	The name of the requirement document (string, optional). Within the XML format there is a separate tag containing the document name.

The column names of the import files are case-insensitive and underscores (“_”) within the names will be ignored.

The only required column or XML attribute of requirements within an import file is the requirement text.

4.1 XML format extensions

The XML format allows importing nested chapter structures. Chapters may contain requirements and text elements. Those elements are distinguished by the XML tag name (i.e. “<requirement>” and “<text>”).

Chapters have the following XML attributes:

XML attribute	Description
name	Text (title) of the chapter
id	(integer number, optional)

All other XML attributes written during export of a requirement document are skipped when found within the import file.

4.2 ID handling

The requirement ID is optional, because you may want to import a pure text document (that has no requirement IDs yet). In this case, the automatic requirement ID assignment of TESSY will be used (as if requirements were added manually).

Another import with the same file will **overwrite** the requirements in the order given within the import file.

IDs may be arbitrary strings, they are **not** restricted to integers.

4.3 Version handling

The requirement version is optional, because you may have no explicit version within imported requirement documents. In this case, the requirement version will be assigned by TESSY automatically (i.e. if a requirement is new or changed, it will get the next available version number for this requirement).

4.4 ReqIF format

TESSY V3.1 supports importing ReqIF (Requirements Interchange Format) formatted files. Because some requirement tools (e.g. DOORS) may not already support the latest ReqIF version, please check for tool updates if the exported ReqIF formatted file cannot be imported by TESSY.

Imported ReqIF documents can be exported again with additional user defined attributes filled with the TESSY test results for all tests linked to the respective requirements. Refer to section 5 for details.

4.5 Examples

The following sections provide short examples for import files. Refer to section 4.6 for a description where to find the DTD file containing the detailed description of the XML import file format.

4.5.1 CSV/TSV

Below is an example for an CSV import file containing the requirement id, text and short description:

```
RequirementID, RequirementText, RequirementShortDescription
42, "Text of the first requirement", "First RQMT"
43, "Text of the second requirement", "Second RQMT"
4711, "Text of the third requirement", "Third RQMT"
13, "Text of the fourth requirement", "Fourth RQMT"
```

4.5.2 XML

Below is an example for an XML import file containing the requirement id, text and short description:

```
<?xml version="1.0" encoding="UTF-8"?>
<documentlist>
  <requirementdocument
    name="Ice Warning Specification"
    description=""
    alias="IWS">
    <requirement
      id="1"
      shortdescription="Introduction"
      text="The ice warning and error indication depends on the
temperature.">
    </requirement>
    <chapter
      name="Environmental conditions">
      <requirement
        id="2"
        shortdescription="Measurement"
        text="The ice warning display shall be based on the
measured outside air temperature.">
      </requirement>
      <requirement
        id="3"
        shortdescription="Temperature range"
        text="The ice warning shall operate for temperatures
from -60&#176;C to +80&#176;C.">
      </requirement>
    </chapter>
  </requirementdocument>
</documentlist>
```

4.6 XML DTD format description

The DTD used for verifying the XML import files is part of the TESSY installation wrapped in the following **.jar** file:

```
Program Files\Razorcat\TESSY_3.0\bin\plugins\com.razorcat.ite.importer.model_1.0.0.jar  
requirement_import.xsd
```

The file can be extracted from the **.jar** archive to review its contents.

5 Exporting ReqIF documents

Exporting of ReqIF formatted requirement documents is only possible when the requirement document has also been imported before in ReqIF format. The purpose of such an export is to export TESSY test results for the requirements. The workflow would be like follows:

1. Import a ReqIF document. The ReqIF document has to contain already the attributes for the expected results. The attribute can be empty or contain already the placeholders for the results.
2. Create tests within TESSY and assign the test cases to the requirements of the imported document.
3. Define the value for the attributes at document level to contain text with placeholders for the results for each requirement.
4. Generate an execution coverage report for your TESSY modules: This will fill the placeholders within your attribute texts with the actual results.
5. Export the ReqIF document which will now contain the additional requirement attributes filled with the actual test results.

In this way you can import the test results assigned to your requirements back into your requirements management system.

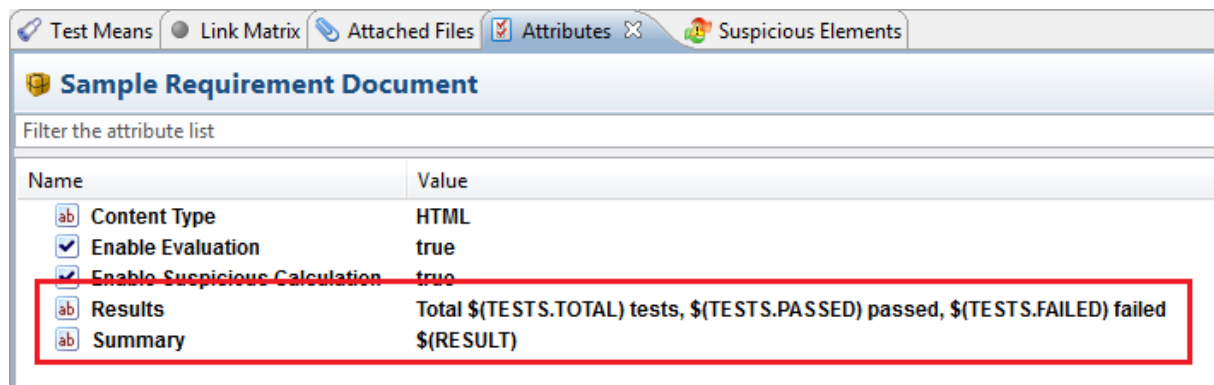
Please note: For technical reasons it is necessary to commit the requirement document after a successful import and then run the import again to add the imported ReqIF file as attached file to the requirement document (Files can only be attached to committed documents). This original ReqIF file will be the basis for the export operation: Any contents added or changed within the requirement document will be merged into the originally imported ReqIF file.

5.1 Creating additional result attributes

You can create any number of attributes for requirement documents containing special placeholders as listed within the table below. The placeholders represent test results related information for a single requirement.

Placeholder name	Content
\$(RESULT)	Contains the status text as used within the execution coverage report
\$(TESTS.TOTAL).	Contains the number of test cases linked to the respective requirement
\$(TESTS.PASSED)	Contains the number of passed test cases
\$(TESTS.FAILED)	Contains the number of failed test cases

An example for such attributes is shown below (attributes at the document level):



Name	Value
<input type="checkbox"/> Content Type	HTML
<input checked="" type="checkbox"/> Enable Evaluation	true
<input checked="" type="checkbox"/> Enable Suspicious Calculation	true
<input type="checkbox"/> Results	Total \$(TESTS.TOTAL) tests, \$(TESTS.PASSED) passed, \$(TESTS.FAILED) failed
<input type="checkbox"/> Summary	\$(RESULT)

You can create as many different attributes as you need.

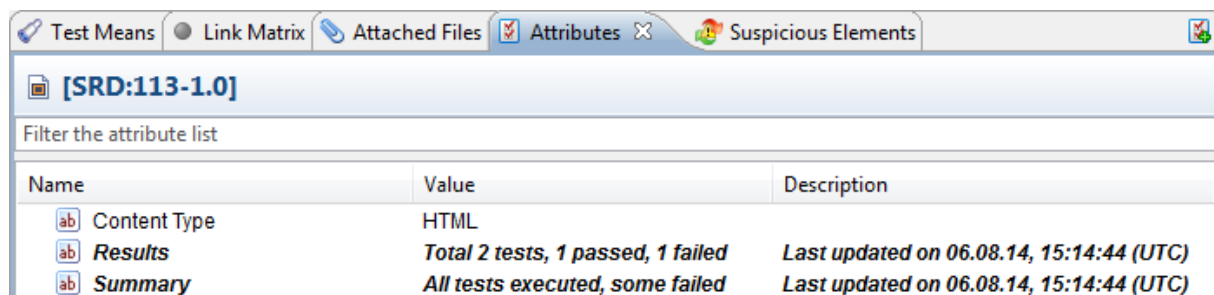
5.2 Updating the result attributes

After executing tests with test cases that are linked to some of the requirements of the given document, you need to run an execution coverage report. The placeholders within the attribute text will automatically be replaced by actual information as listed within the execution coverage report for each requirement:

Coverage Overview Arranged by Requirement

Identifier Text	State	Number of Tests
[SRD:112-1.0] First requirement	All Tests passed	1
[SRD:113-1.0] Second requirement	All Tests executed, some failed	2
[SRD:114-1.0] Third requirement	No Tests available	0

The attributes for the second requirement would be filled with the text shown below:



Name	Value	Description
Content Type	HTML	
Results	Total 2 tests, 1 passed, 1 failed	Last updated on 06.08.14, 15:14:44 (UTC)
Summary	All tests executed, some failed	Last updated on 06.08.14, 15:14:44 (UTC)

The description column contains the time and date of the last automatic update of the attribute text (triggered by generating the execution coverage report).