Import TestCases into TestLink

Purpose of the tool

If you want to create a lot of testcases in TestLink easily, the easiest way is to use tables.

TestLink is able to use XML to import testcases. This document explains how to import Testcases into TestLink using XML files. Excel is used as editor to add new testcases.

Version information

- **Excel Filename** : 02-ImportTestCasesIntoTestLink.xls
- **Applicable Excel Macro Version :** 20150831
- **Works with :**
  - ✓ TestLink 1.9.3 ➔ TestLink 1.9.13

Summary

Testcases declaration using Excel

To declare testcases, the excel pattern named “02-ImportTestCasesIntoTestLink.xls” must be used. This file could be used to update existing tests or to create new tests.

Excel pattern overview

Test Suite		Import TestCases From TestLink	Test Case	Export TestCases To TestLink	Steps		Coverage
Name	Details	Name	Summary	Step	Expected Result		Document ID
Parent	Details about the parent test suite						
myTestSuite	Details about my TestSuite						
		A first TestCase	The summary of the First TestCase	The first Step	Expected result for the first step		Req-SandBox-1
				A second Step	And an expected result for the second step		
		A second TestCase	A second test Case for My TestSuite	The first Step	Expected result for the first step		Req-SandBox-3
				A second Step	And an expected result for the second step		
Parent	Details about the parent test suite						
YourTestSuite	Details about your TestSuite						
		Your first TestCase	The summary of the First TestCase of your testSuite	The first Step	Expected result for the first step		Req-SandBox-4
				A second Step	And an expected result for the second step		
		Your second TestCase	A second test Case for Your TestSuite	The first Step	Expected result for the first step		Req-SandBox-4; Req-SandBox-2
				A second Step	And an expected result for the second step		
				A Third Step	Expected result for the third step		

Figure 1 : Excel pattern information overview

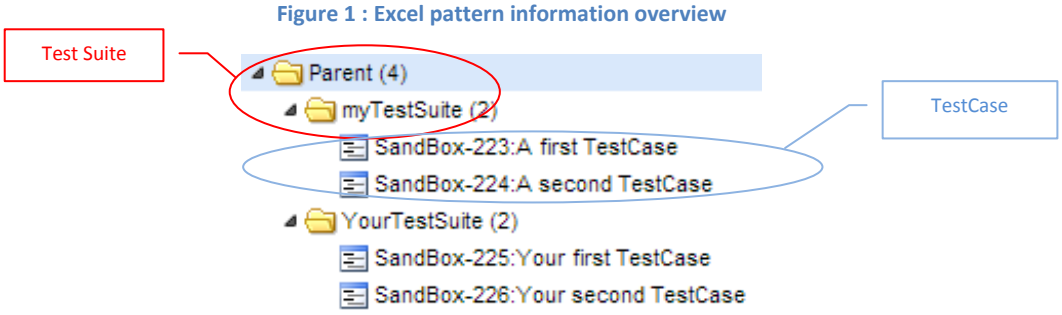

Figure 2 : imported results overview

- The “Test Suite” section (dark blue columns header) can be used to declare folders trees in TestLink (folders are named “Test Suite Operations” in TestLink GUI)
- The “Test Case” section (cyan columns header) can be used to declare Test Cases in TestLink (Test Cases are named “Test Case Operations” in TestLink GUI)
- The “Custom Fields” section (dark pink columns header) is hidden by default. This section can be used to declare several Custom Fields in TestLink, by adding new columns if needed
- The “Steps” section (yellow columns header) can be used to declare Test Cases Steps and expected results in TestLink
- The “Coverage” section (grey columns header) can be used to declare Test Case coverage in TestLink
- The “Time Management” section (Orange columns header) is hidden by default. This section can be used to declare the Estimated execution Duration of a TestCase
- The “Test Suite” section and the association of the “Test Case”, “Custom Fields”, “Steps”, “Coverage” and “Time Management” sections **must not** be filled on the same line
- All “Test Case” declared after a “Test Suite” section will be inserted in the most recent “Test Suite” defined

- "TestCase", "Custom Fields", "Coverage" and "Time Management" sections of each Testcase **must** be provided only once, even if the Testcase is multi-steps
- Consecutives lines in the "Test Suite" section are considered as a "subfolders" (tree) declaration.
- The "Test Suite" **must provide the path from the root directory** each time a Test Suite is declared in the file. Test Suite "Name" and "Details" **must be** declared several times in the same file if the same folder is used for several sub-folders (see "Parent" example in figures above).

Excel columns content explanations

(Red = mandatory, Black = Optional)

- **Columns used for a Test Suite declaration**
  - **Name** : TestSuite Name
  - **Details** : TestSuite Description
- **Columns used for a TestCase declaration**
  - **TC#** : TestCase Number. If filled, this number is used during import to determine the node order of the test in the TestSuite. If not filled, automatically generated by TestLink during import.
  - **Name** : TestCase name
  - **Summary** : TestCase short description
  - **PreCondition** : Testcase Preconditions
  - **ExecutionType** : Testcase Execution type

Value in TestLink	Manual	Automatic
Value to use in Excel	1	2

  - **Importance** : Testcase Importance

Value in TestLink	Low	Medium	High
Value to use in Excel	1	2	3

- **Columns used for Custom Fields declaration**
  - **<Custom1>** : User defined field
  - **<Custom2>** : User defined field
- **Columns used for Steps declaration**
  - **Step** : What to do (action). One Excel row per step
  - **Expected Result** : Expected result for each step. One Excel row per expected result.
  - **StepExecutionType** : Step Execution type. Optional in Excel file, Mandatory in XML generated file. Forced to 1 (Manual) in XML file if not filled in Excel file.

Value in TestLink	Manual	Automatic
Value to use in Excel	1	2

- **Columns used for Requirements Coverage declaration**

  - **Spec Title** : Requirement Title that is covered by this Testcase.
  - **Document ID** : Requirement Doc-ID(s) that is/are covered by this Testcase. To cover more than one Doc-ID, separator is semi-comma character (;)

- **Columns used for Time Management declaration**

  - **Estimated exec. Duration** : The estimated duration to execute the test. In Minutes.

Excel text length validation

Some TestLink fields are length limited. Each limit is configured in Excel file to inform the user when the limit is reached. When the limit is reached, the XML generation still works, but TestLink will truncate the value during import.

Each cell that contains too many characters is displayed with a red background.

- Requirement Specification Operation Doc ID (TestSuite Name) : 75 characters maximum
- Requirement Specification Operation Doc Title (TesctCase Name) : 100 characters maximum

Test Suite		TestCases Generation	Test Case		
Name	Details	Name	Summary	PreConditions	
Parent	Details about the parent test suite				
myTestSuite where the name is very very very very very very long so the limit is reached	Details about my TestSuite				
		A first TestCase	The summary of the First TestCase		
		A second TestCase where the name is very very very very very very very very very very long so the limit is reached	A second test Case for My TestSuite	A precondition is defined	
Parent	Details about the parent test suite				
YourTestSuite	Details about your TestSuite				
		Your first TestCase	The summary of the First TestCase of your testSuite	Your precondition	
		Your second TestCase	A second test Case for Your TestSuite		

Figure 3 : Excel text length validation

XML file generation

When all Testcases has been written in Excel, just click on 'Export TestCases To TestLink' button to execute the Excel Macro. Of course, macro execution must be allowed for the '02-ImportTestCasesIntoTestLink.xls' file.

If there is a problem with the content, the popup should explain why the generation has failed.

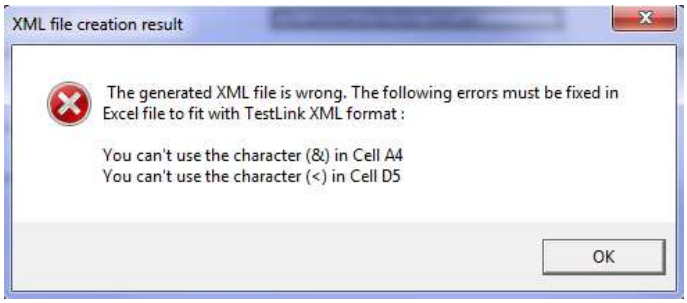

Figure 4 : XML file generation result KO

A log file containing all errors can be created if needed.

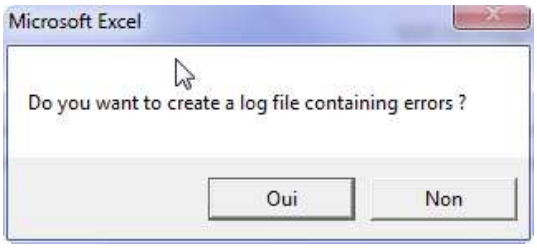

Figure 5 : Save the log file

The log file is created in the same Excel file folder, with the Excel file name (with a .log suffix).

If everything is OK with the file content, a popup will appear with the following text displayed

Figure 6 : XML file generation result OK

The XML file is created in the same Excel file folder, with the Excel file name (the .xls suffix is replaced by a .xml suffix).

When the XML generation is done, a popup ask if the requirement coverage must be done

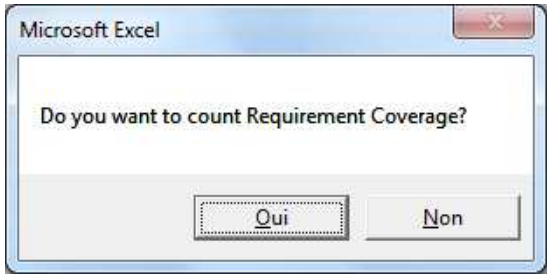

Figure 7 : Requirement Coverage question

If chosen, a new sheet is displayed, showing how many occurrence of each requirement is present

Requirements List	Requirements without duplicates	Occurrences
Req-SandBox-1	Req-SandBox-1	1
Req-SandBox-3	Req-SandBox-3	1
Req-SandBox-4	Req-SandBox-4	2
Req-SandBox-4		
Req-SandBox-2		

Figure 8 : Requirement Coverage result

This information should be useful to update the file “01-ImportRequirementsIntoTestLink” which has to provide the number of testcases needed to cover the requirement.

TestCases Import into TestLink

XML file generated by the macro must be used during the import process.

1) Select the folder representing the root directory where all “Test Suites” and “TestCases” will be inserted. This can be the root directory or a folder in the Test Suites tree.

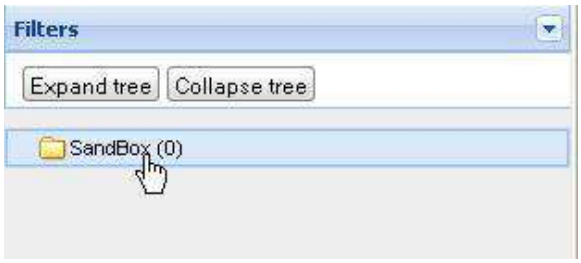

Figure 9 : select destination folder for requirements import

2) Select the import button

Figure 10 : Select import menu when the root directory is selected

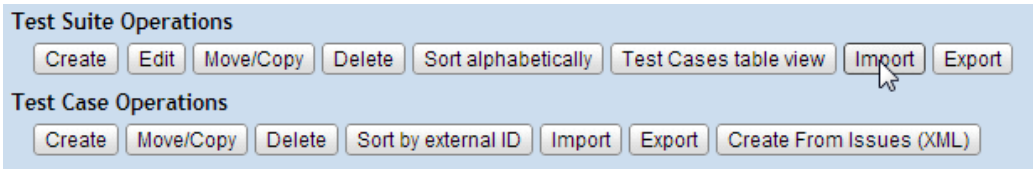

Figure 11 : Select import menu when a Test Suite (sub-folder) is selected

3) Configure the import options

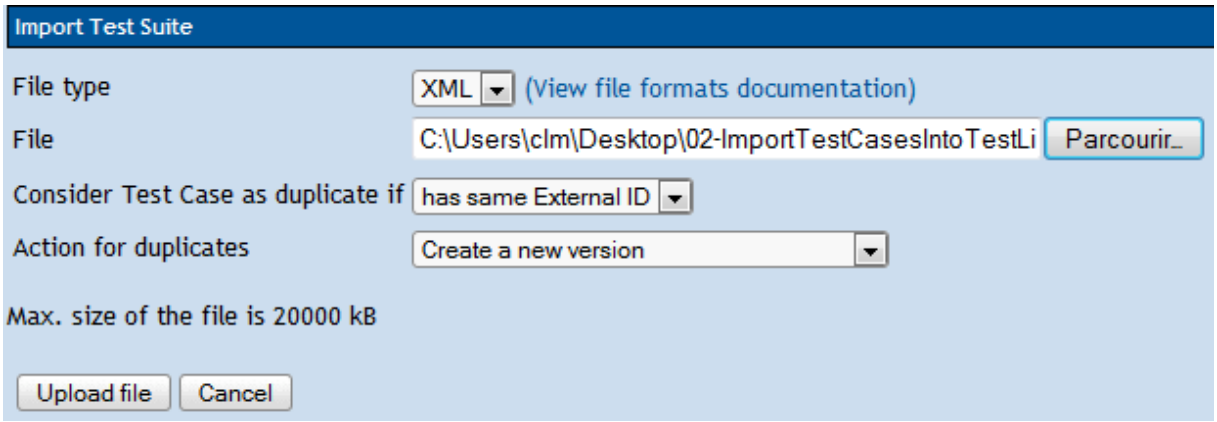

Figure 12 : TestCases import options

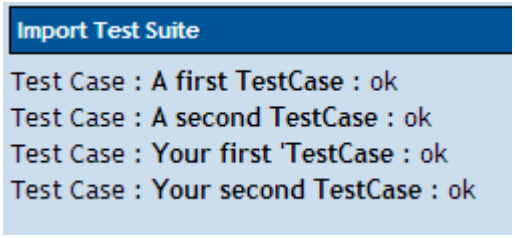

Figure 13 : import Testcases result

Import into Excel TestCases Exported from TestLink

To export from TestLink, use the Export Function provided by TestLink :

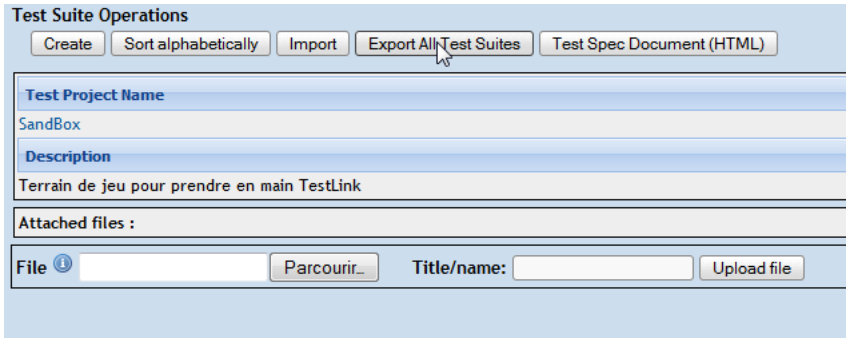

Figure 14 : TestCases Export function

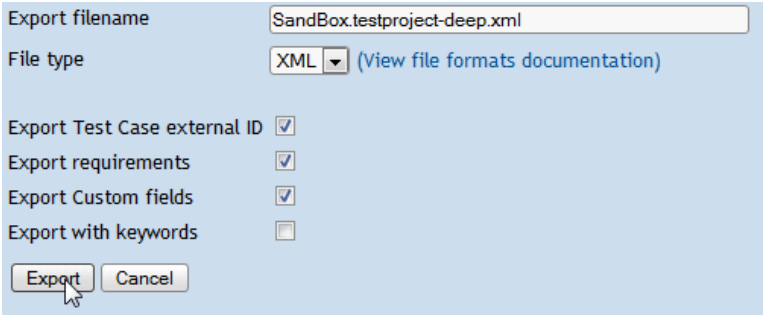

Figure 15 : TestCases Export options

WARNING :

TestCase content is depending of the text editor that you use when you write TestCases (CKEditor, FCKEditor, …). The provided function should be used to import testcases ONLY WHEN NO ADVANCED EDITOR IS USED :

```
$tlCfg->gui->text_editor['design'] = array( 'type' => 'none');
$tlCfg->gui->text_editor['steps_design'] = array( 'type' => 'none');
```

Editor internal information (HTML tags) will be showed if an advanced editor has been used, and TestCases content could be altered when importing back to TestLink.

If you want to import TestCases exported from TestLink, just click on 'Import TestCases From TestLink' button to execute the Excel Macro. Of course, macro execution must be allowed for the '02-ImportTestCasesIntoTestLink.xls' file.

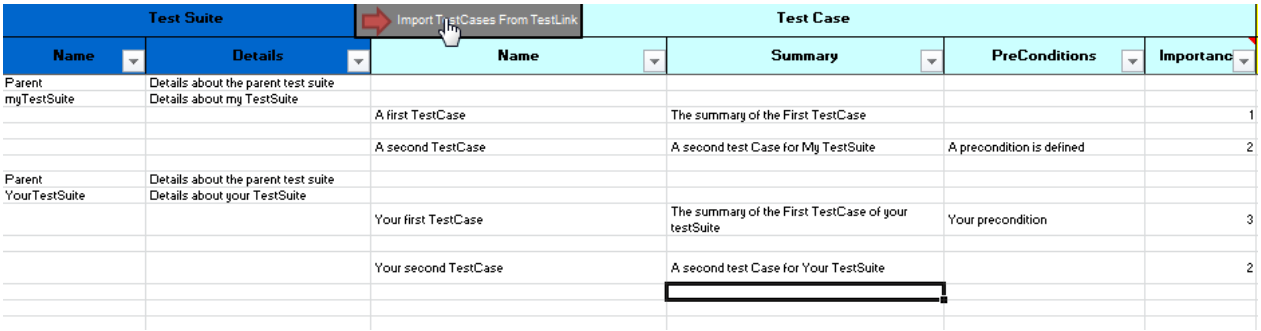

Figure 16 : TestCases Export options

A confirmation message is displayed; the user must acknowledge that all current data in the spreedsheet will be lost.

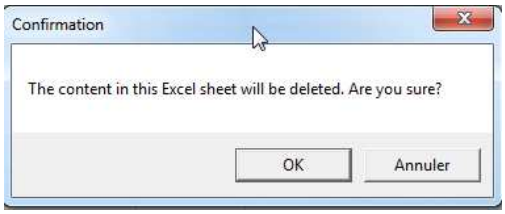

Figure 17 : Confirmation message

The file generated by TestLink during export must be provided.

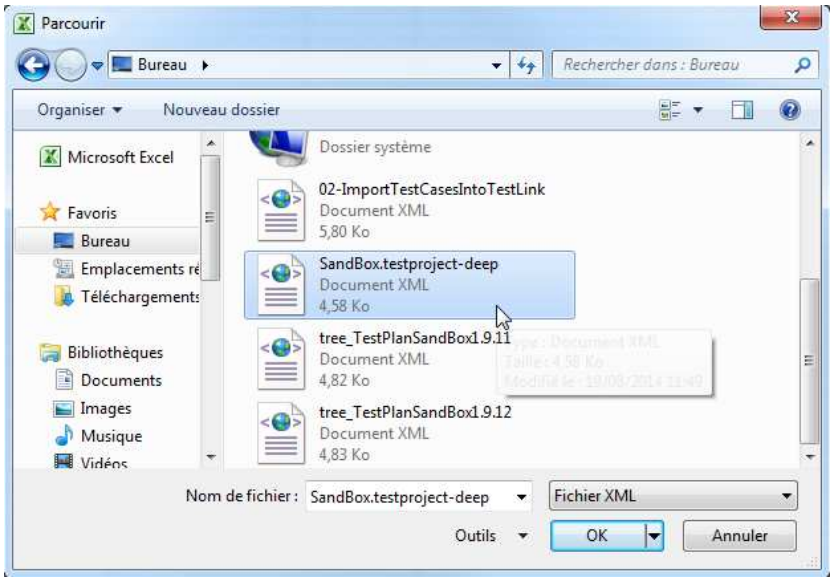

Figure 18 : XML File Selection

The file generated by TestLink is parsed by the Excel macro. Depending of the content, this analyze can be time-consuming. Excel doesn't answer during this step. Just wait that Excel give back the hand. When finished, the Excel content represents the content of each testcase.

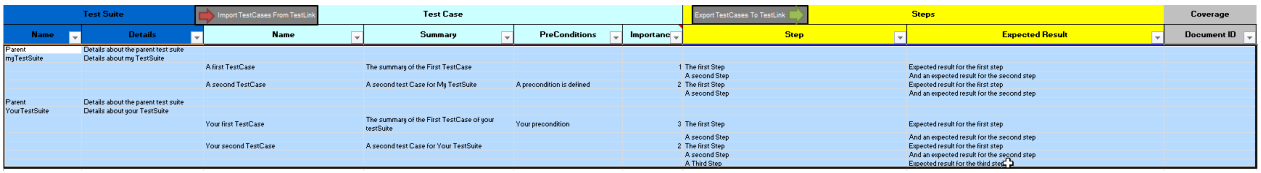

Figure 19 : Result after Import into Excel

You can now edit testcases and import content back to TestLink to create a new version/revision.

After an export, the External ID is correct in the Excel file, the best configuration to Import back into testlink is to use External ID as identifier. With this provided configuration, you can modify the TestCase name if needed :

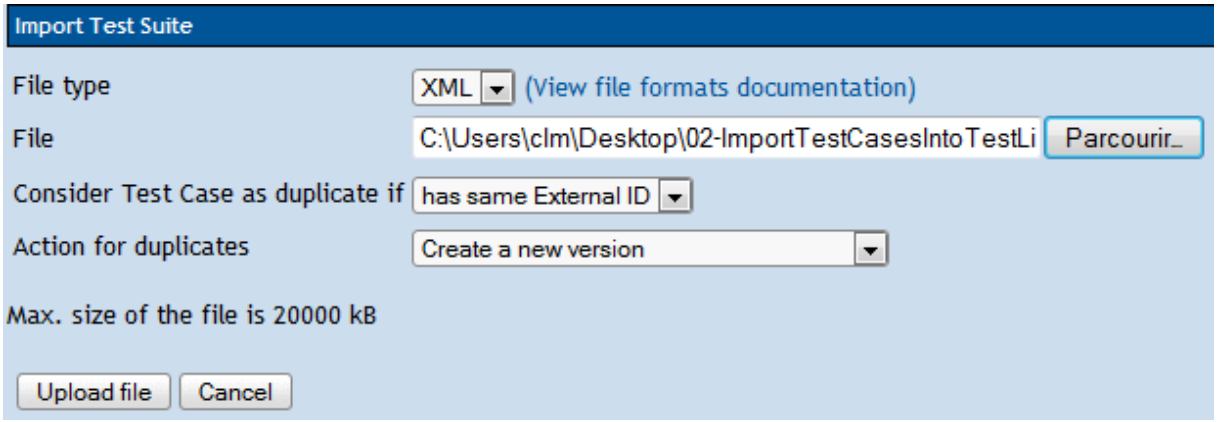

Figure 20 : Import TestCases update – preconized configuration

Tips and problems

- By default, **columns representing optional information** are hidden in Excel file
- In **Coverage Document ID column**, if more than one Requirement Doc-ID is provided, each Doc-ID can be provided on its own row (Alt + Enter to add a Carriage Return in the cell), separator between each requirement is semi-comma (;)
- **Generate a Testcases XML file** with Excel has been tested with Excel 2003, Excel 2007 and 2010 versions.
- **XML Test Suite** haven't to be created in TestLink before import
- If your **Testcases titles** are longer than 100 characters, titles will be reduced by TestLink during import. You have to modify const.inc.php to reconsider the value of testcase title field size (max is 255 caracters in database)
- If your **XML Testcases file** is bigger than 400Kb, you have to edit custom_cfg.inc.php to change "$tlCfg->import_file_max_size_bytes" parameter. To set new limit file size to 1Mb, modify this parameter from '409600' to '1024000'.
- The **XMLTestcases file encoding** depends of Excel version. In French version, the Excel generated file is not UTF-8 but ANSI. Encodage information is set to "ISO-8859-1" in XML file, allowing the use of Latin characters (like é, è or à used in French). If a "XML Loading Failed" is displayed during XML import in TestLink, check the file encoding.
- During **import into TestLink**, TestLink doesn't accept that two testcases have the same name in the same TestSuite.
- In **import result log view**, you can search for "ok", "skipped", "updated" or "duplicate" keywords to verify that TestLink Testcases import comportment is coherent with your needs.
- In **import from TestLink**, node hierarchy deep is configured with a maximum of 10 "sub-nodes". If value must be adapted, declarations in function "ParseXMLTestCases" (TestSuiteName and TestSuiteDetails tabs) should be modified

Document Changelog

Excel File Version	Major Changes
20120314	Initial Release
20130218	Explain the Requirement Coverage function Import has been successfully tested within TestLink 1.9.3, 1.9.4, 1.9.5 and 1.9.6
20130530	Explain the Requirement Coverage function Import has been successfully tested within TestLink 1.9.7
20130916	Update How To information
20140129	Update How To information
20140403	Update How To information to explain the use of sub-folders tree solution (multi TestSuites declaration)
20140415	Update How To information to explain how Excel is used to validate text length Import has been successfully tested within TestLink 1.9.8 and 1.9.9 Add explanations about the new XML tag <estimated_execution_duration>
20140724	Update How To information. Import has been successfully tested within TestLink 1.9.10

Excel File Version	Major Changes
20140827	Update How To information to explain how to use the “Import TestCases From TestLink” button. Import has been successfully tested within TestLink 1.9.11
20141214	Update How To information to fit macro comportment. Import has been successfully tested within TestLink 1.9.12
20150331	Bug correction : Max sub-node hierarchy is now configured to 10 (complex node hierarchy). Import has been successfully tested within TestLink 1.9.13
20150831	Bug correction : first requirement is displayed twice in requirement counter