

Import / export file formats

TestLink version 1.9

Author: Francisco Mancardi
Version: 1.0
Status: Updated for TL 1.9

© 2004 - 2010 TestLink Community

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. The license is available in ["GNU Free Documentation License" homepage](#).

Revision History

#	Description	Date	Author
0.1	Initial document	20070728	Francisco Mancardi
0.2	Added XLS format for test cases. Code contributed by lightbulb Added XML format for results	20071101	Francisco Mancardi
0.3	Notes about internal and external ID New tag supported for results on TL 1.8	20080911	Francisco Mancardi
0.4	Test Case XML – added support for custom fields	20090106	Francisco Mancardi
0.5	Test Case XML – added support for link requirements	20090207	Francisco Mancardi
0.6	Updated Format for version 1.9	20100227	Francisco Mancardi
1.0	Updated documentation layout, corrected license, merged information from User manual (docbook support,etc.)	04/03/10	Martin Havlat

1 Introduction

TestLink is web based **Test Management tool**. This manual describes a format of files for import and export data.

See the **User manual** and **Installation manual** for more information about tool. The latest documentation is available on web <http://www.teamst.org> .Feel free to use our [forum](#) if you have questions that the manual doesn't answer.

Table of Contents

1	Introduction	3
2	Import and Export data	4
2.1	Export/Import Test Project	4
2.2	Import/Export Test suite	5
2.3	Just one Test Case	6
2.4	All Test Cases in test suite	9
2.5	Import/Export Keywords	9
2.6	Import/Export Software Requirements	10
2.7	Results import	12
2.8	Import Test Cases from Excel via XML	12
2.9	Platforms	14
2.10	Custom Fields	16

2 Import and Export data

TestLink supports several ways to share data. See the next table for overview. In addition you can consider to use TestLink API to deliver supported data.

There is amount of file examples in directory `testlink/docs/file_examples/`.

Troubleshooting: No answer for import action? Check size of imported file. There are limits in TestLink configuration and web server settings. Check if DOM module is loaded for your web server.

Item	File format	Import	Export	Notes
Test project	XML	X	X	All test suites and test cases. You can choose if export also assigned keywords.
Test suite	XML	X	X	Test suite details, All test cases and child test suites and test cases. You can choose if export assigned keywords.
Test case	XML	X	X	Two types of exports can be done: <ul style="list-style-type: none">Just one test caseAll test cases in test suite. Custom Fields and Requirements assigned are exported. Keywords export is optional.
Test case	XLS	X		Keywords import is NOT supported.
Keyword	CSV, XML	X	X	All test project's keywords
Requirement	CSV, XML	X	X	
Requirement	CSV DOORS, DocBook	X		
Results	XML	X		
Platforms	XML	X	X	New on 1.9
Custom Fields	XML	X	X	New on 1.9

Table 1: Items that can be exported/imported

Limitation: Attached files and custom fields¹ are not imported/exported.

Table format (CSV) is not directly supported in some cases. You should convert it into XML before import. See below for more.

Definition for **Internal** and **Documentation Identifiers**

- Every object has its internal ID , this ID is value of ID column in database table
- Test cases and requirements are special case because they have internal and document ID.
- Every time you see keyword ID in xml format it indicates INTERNAL ID.

2.1 All Test Project

User can import or export Test Project including Description of the project, Test Specification and Keywords. The next two pictures show tree menu with data and the same data as XML file.

Warning: You can reach a server memory limit for larger amount of Test cases.

¹ CF except Test cases.

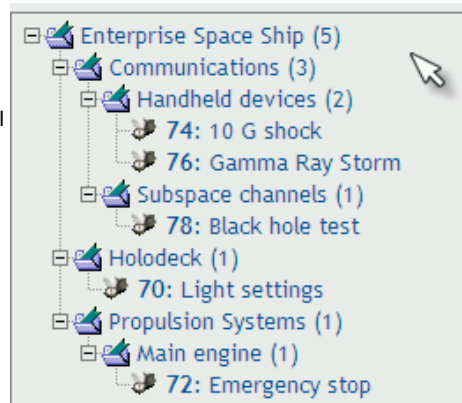
```

<?xml version="1.0" encoding="UTF !"?"
<tests#ite name=""
  <details"<$%&' (T(%)) "</details"
  <tests#ite name="omm#nications"
    <details"<$%&' (T(%<p"&omm#nication **stems of all
t+pes</p"))"</details"
    <tests#ite name=",and -eld devices"
      <details"<$%&' (T(%)) "</details"
      <testcase name="10 . s-ock"
        <s#mmar+"<$%&' (T(%)) "</s#mmar+"
        <steps"<$%&' (T(%)) "</steps"
        <expectedres#lts"<$
%&' (T(%)) "</expectedres#lts"
      </testcase"
      <testcase name=".amma /a+ *torm"
        <s#mmar+"<$%&' (T(%)) "</s#mmar+"
        <steps"<$%&' (T(%)) "</steps"
        <expectedres#lts"<$%&' (T(%)) "</expectedres#lts"
      </testcase"
    </tests#ite"
  <tests#ite name="*#0space c-annels"
    <details"<$%&' (T(%<p"1nl+ 0asic s#0space feat#res</p"))"</details"
    <testcase name="2lack -ole test"
      <s#mmar+"<$%&' (T(%)) "</s#mmar+"
      <steps"<$%&' (T(%)) "</steps"
      <expectedres#lts"<$%&' (T(%)) "</expectedres#lts"
    </testcase"
  </tests#ite"
</tests#ite"

<tests#ite name=",olodeck"
  <details"<$%&' (T(%)) "</details"
  <testcase name="3ig-t settings"
    <s#mmar+"<$%&' (T(%)) "</s#mmar+"
    <steps"<$%&' (T(%)) "</steps"
    <expectedres#lts"<$%&' (T(%)) "</expectedres#lts"
  </testcase"
</tests#ite"

<tests#ite name="4rop#lsion **stems"
  <details"<$%&' (T(%)) "</details"
  <tests#ite name="5ain engine"
    <details"<$%&' (T(%)) "</details"
    <testcase name="6mergenc+ stop"
      <s#mmar+"<$%&' (T(%)) "</s#mmar+"
      <steps"<$%&' (T(%)) "</steps"
      <expectedres#lts"<$%&' (T(%)) "</expectedres#lts"
    </testcase"
  </tests#ite"
</tests#ite"
</tests#ite"

```



2.2 Import/Export Test suite

XML Example – Test Suite without keywords

```

<?xml version="1.0" encoding="UTF !"?"
<tests#ite name=",and -eld devices"
  <details"<$%&' (T(%)) "</details"
  <testcase name="10 . s-ock"
    <s#mmar+"<$%&' (T(%)) "</s#mmar+"
    <steps"<$%&' (T(%)) "</steps"
    <expectedres#lts"<$%&' (T(%)) "</expectedres#lts"
  </testcase"
  <testcase name=".amma /a+ *torm"

```



```

    <s#mmar+ "<$%&' (T(%)) "</s#mmar+ "
    <steps "<$%&' (T(%)) "</steps "
    <expectedres#lts "<$%&' (T(%)) "</expectedres#lts "
  </testcase "
</tests#ite "

```

XML format example: Test Suite with keywords

```

<?xml version="1.0" encoding="UTF !"?"
<tests#ite name=" ,and -eld devices " "
  <details "<$%&' (T(%)) "</details "
  <testcase name="10 . s-ock " "
    <s#mmar+ "<$%&' (T(%)) "</s#mmar+ "
    <steps "<$%&' (T(%)) "</steps "
    <expectedres#lts "<$%&' (T(%)) "</expectedres#lts "
    <keywords>
      <ke+7ord name="8l+ngon " "
        <notes "<$%&' (T(%8l+ngon ke+7ord notes)) "</notes "
      </ke+7ord "
    </ke+7ords "
  </testcase "
  <testcase name=" .amma /a+ *torm " "
    <s#mmar+ "<$%&' (T(%)) "</s#mmar+ "
    <steps "<$%&' (T(%)) "</steps "
    <expectedres#lts "<$%&' (T(%)) "</expectedres#lts "
    <keywords>
      <ke+7ord name="8l+ngon " "
        <notes "<$%&' (T(%8l+ngon ke+7ord notes)) "</notes "
      </ke+7ord "
      <ke+7ord name="5oon rocks " "
        <notes "<$%&' (T(%5oon rocks ke+7ord notes)) "</notes "
      </ke+7ord "
    </ke+7ords "
  </testcase "
</tests#ite "

```

2.3 Just one Test Case

ID 78 :: Test Case Black hole test									
Version 1									
Summary									
This procedure must be done once a week, with this safety device disabled:									
1. X45HH 2. YY89-000-JI									
Steps	Expected Results								
Preset bias to 0	<table border="1"> <thead> <tr> <th colspan="2">Main Results</th> </tr> </thead> <tbody> <tr> <td>Spin value</td> <td>9.9</td> </tr> <tr> <td>Opposite Angle</td> <td>18 rad</td> </tr> <tr> <td> </td> <td> </td> </tr> </tbody> </table>	Main Results		Spin value	9.9	Opposite Angle	18 rad		
Main Results									
Spin value		9.9							
Opposite Angle	18 rad								
Enable long range communications control									
Simulate black hole interference									
Keywords: Moon rocks									
Created on 27/07/2007 15:16:52 by admin									
Last modified on 27/07/2007 16:16:33 by admin									

Example of XML file: Test case with keyword

```
<?xml version="1.0" encoding="UTF !"?"
<testcases"
  <testcase name="2lack -ole test"
    <s#mmar+"
      <![CDATA[<p>This procedure must be done once a week, with this safety device
disabled:</p>
      <ol"<li"9:;,,</li"<li"<li"<!= 000 >?</li"</ol""))"
    </s#mmar+"
    <steps><![CDATA[
      <p"4reset 0ias to 0</p"
      <p"6na0le <strong"long range</strong" comm#nications control</p"
      <p>Simulate black hole interference</p>]]> </steps"
    <expectedres#lts><![CDATA[
<ta0le 7idt-="@00" cellspacing="1" cellpadding="1" 0order="1"
  <caption"5ain /es#lts</caption"
  <t0od+"
    <tr"<td"*pin val#e</td"<td"=.=</td"</tr"
    <tr"<td"lpposite (ngle</td"<td"1! rad</td"</tr"
    <tr"<td"<td"</td"<td"</td"</tr"
  </t0od+"
</ta0le""))"
</expectedres#lts"
  <ke+7ords"
    <ke+7ord name="5oon rocks"
      <notes"<$%&'(T(%5oon rocks ke+7ord notes))"</notes"
    </ke+7ord"
  </ke+7ords"
</testcase"
</testcases"
```

Example : XML – Test Case with custom fields

```
<?xml version="1.0" encoding="UTF !"?"
<testcases"
  <testcase name="2lack ,a7k test"
    <s#mmar+"
      <$%&'(T(%p"T-is proced#re m#st 0e done once a 7eekC 7it- t-is safet+ device
disa0led</p"
      <ol"<li"9:;,,</li"<li"<li"<!= 000 >?</li"</ol""))"
    </s#mmar+"
  </testcase"
</testcases"
```

```

</s#mmar+"
<steps"<$%&' (T(%
  <p"4reset 0ias to 0</p"
  <p"6na0le <strong"long range</strong" comm#nications control</p"
  <p"*im#late 0lack -ole interference</p"))"
</steps"
<expectedres#lts"<$%&' (T(%
  <ta0le 7idt-="@00" cellspacing="1" cellpadding="1" border="1"
  <caption"5ain /es#lts</caption"
  <t0od+"
    <tr"<td"*pin val#e</td"<td"=. =</td"</tr"
    <tr"<td"1pposite (ngle</td"<td"1! rad</td"</tr"
    <tr"<td"An0spB</td"<td"An0spB</td"</tr"
  </t0od+"
  </ta0le"))"
</expectedres#lts"
<c#stom_fields"
  <c#stom_field"
    <name"<$%&' (T(%&F_*8?33*_E66'6'))"</name"
    <val#e"<$%&' (T(%F( 6ngineer))"</val#e"
  </c#stom_field"
  <c#stom_field"
    <name"<$%&' (T(%&F_6*T?5(T6'_696&_T?56))"</name"
    <val#e"<$%&' (T(%1@))"</val#e"
  </c#stom_field"
</c#stom_fields"
</testcase"
</testcases"

```

Example: XML – Test Case with requirements

```

<?xml version="1.0" encoding="UTF !"?"
<testcases"
<testcase internalid="1@G::" name=",ig- speed"
  <node_order"<$%&' (T(%0))"</node_order"
  <externalid"<$%&' (T(%1!@))"</externalid"
  <s#mmar+"<$%&' (T(%))"</s#mmar+"
  <steps"<$%&' (T(%))"</steps"
  <expectedres#lts"<$%&' (T(%))"</expectedres#lts"
  <reH#irements"
    <reH#irement"
      <reH_spec_title"<$%&' (T(%/*46& 001))"</reH_spec_title"
      <doc_id"<$%&' (T(%6E. 000@))"</doc_id"
      <title"<$%&' (T(%5ain 'eflector))"</title"
    </reH#irement"
    <reH#irement"
      <reH_spec_title"<$%&' (T(%/*46& 001))"</reH_spec_title"
      <doc_id"<$%&' (T(%'1& 00=))"</doc_id"
      <title"<$%&' (T(%>ames 2ond))"</title"
    </reH#irement"
  </reH#irements"
</testcase"

<testcase internalid="1@G:G" name=",alf speed stop"
  <node_order"<$%&' (T(%0))"</node_order"
  <externalid"<$%&' (T(%1!I))"</externalid"
  <s#mmar+"<$%&' (T(%))"</s#mmar+"
  <steps"<$%&' (T(%))"</steps"
  <expectedres#lts"<$%&' (T(%))"</expectedres#lts"
  <reH#irements"
    <reH#irement"
      <reH_spec_title"<$%&' (T(%/*46& 001))"</reH_spec_title"
      <doc_id"<$%&' (T(%6E. 000@))"</doc_id"
      <title"<$%&' (T(%5ain 'eflector))"</title"
    </reH#irement"
    <reH#irement"
      <reH_spec_title"<$%&' (T(%/*46& 001))"</reH_spec_title"
      <doc_id"<$%&' (T(%'1& 00=))"</doc_id"

```



```

        <title"<$%&' (T(%>ames 2ond)) "</title"
    </reH#irement"
</reH#irements"
</testcase"

<testcase internalid="1@G:!" name="#mp start"
    <node_order"<$%&' (T(%0)) "</node_order"
    <externalid"<$%&' (T(%1!:) "</externalid"
    <s#mmar+"<$%&' (T(%)) "</s#mmar+"
    <steps"<$%&' (T(%)) "</steps"
    <expectedres#lts"<$%&' (T(%)) "</expectedres#lts"
    <reH#irements"
        <reH#irement"
            <reH_spec_title"<$%&' (T(%/*46& 001)) "</reH_spec_title"
            <doc_id"<$%&' (T(%6E. 000@)) "</doc_id"
            <title"<$%&' (T(%5ain 'eflector)) "</title"
        </reH#irement"
        <reH#irement"
            <reH_spec_title"<$%&' (T(%/*46& 001)) "</reH_spec_title"
            <doc_id"<$%&' (T(%'1& 00=)) "</doc_id"
            <title"<$%&' (T(%>ames 2ond)) "</title"
        </reH#irement"
    </reH#irements"
</testcase"
</testcases"

```

2.4 All Test Cases in test suite



```

<?xml version="1.0" encoding="UTF !"?"
<testcases"
  <testcase name="10 . s-ock"
    <s#mmar+"<$%&' (T(%)) "</s#mmar+"
    <steps"<$%&' (T(%)) "</steps"
    <expectedres#lts"<$%&' (T(%)) "</expectedres#lts"
  </testcase"
  <testcase name=".amma /a+ *torm"
    <s#mmar+"<$%&' (T(%)) "</s#mmar+"
    <steps"<$%&' (T(%)) "</steps"
    <expectedres#lts"<$%&' (T(%)) "</expectedres#lts"
  </testcase"
</testcases"

```

Test cases in XLS format

Every row must have four columns:

Column number	Contents
1	Test case name
2	summary
3	steps

4	Expected results
---	------------------

First row will be skipped, because is supposed it contains column descriptions.

Example:

Name	Summary	Steps	Expected results
Engine fast start up	Start up on 5 second	Too fast write steps	Kind nothing
Engine emergency stop	Engine stop due to panic button.	<ol style="list-style-type: none"> 1. Unlock panic button 2. Press panic button 3. Press confirm 	Engine must stop right now
Etc.	Etc.	Etc.	Etc.

2.5 Import/Export Keywords

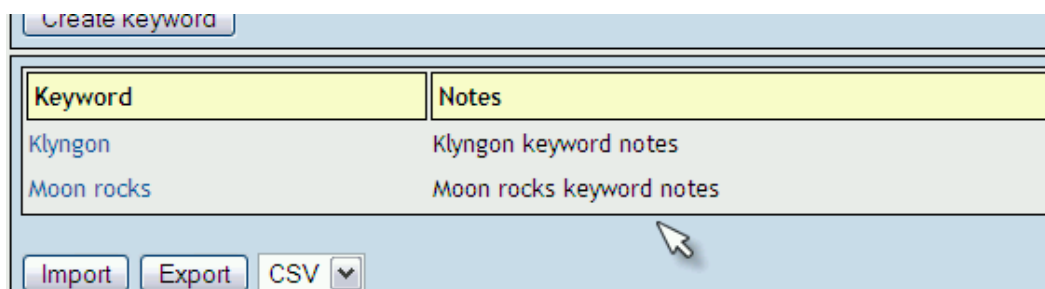


Illustration 1: Keywords frame includes buttons for import and export

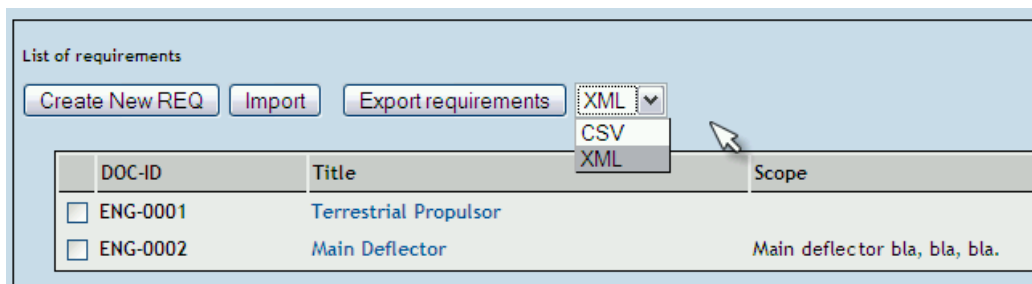
Example of CSV "Keyword;Notes":

```
8l+ngonB8l+ngon ke+7ord notes
5oon rocksB5oon rocks ke+7ord notes
```

Example of XML with keywords:

```
<?xml version="1.0" encoding="UTF !"?"
<ke+7ords"
  <ke+7ord name="8l+ngon" "
    <notes"
      <$%&'(T(%8l+ngon ke+7ord notes))"
    </notes"
  </ke+7ord"
  <ke+7ord name="5oon rocks" "
    <notes"
      <$%&'(T(%5oon rocks ke+7ord notes))"
    </notes"
  </ke+7ord"
</ke+7ords"
```

2.6 Import/Export Software Requirements



CSV file includes "Identifier of document, title, description".

Example of CSV file:

```
6E. 0001cTerrestrial 4rop#lsorc
6E. 000@C5ain 'eflectorC"<p"5ain deflector 0lac 0lac 0la.</p"
```

Example of XML file:

```
<?xml version="1.0" encoding="UTF !"?"
<reH#irements"
  <reH#irement"
    <docid><![CDATA[ENG-0001]]></docid>
    <title"<$%&'(T(%Terrestrial 4rop#lsor))"</title"
    <description"<$%&'(T(%))"</description"
  </reH#irement"
  <reH#irement"
    <docid><![CDATA[ENG-0002]]></docid>
    <title"<$%&'(T(%5ain 'eflector))"</title"
    <description"<$%&'(T(%<p"5aindeflector 0lac 0lac 0la.</p"))"</description"
  </reH#irement"
</reH#irements"
```

2.7 Import rich text format requirements via DocBook

There is limited support of import for documents in such formats as is MSWord or openoffice. You can export original document as DocBook (tested with openoffice2 and 3). Choose import button for your SRS in TestLink. Select type "DocBook".

The exported file is XML. Basic element for default settings could be the next:

```
...
<sectI"
  <title"Title</title"
  ...
  <para"escription</para"
  ...
  <orderedlist"
    <listitem"?tem</listitem"
    ...
  </orderedlist"
  ...
  <informaltable"
    <tgroup"
      <t-ea"
        <ro7" ... <entr+"</entr+" ... </ro7"
      </t-ea"
      <t0od+"
        <ro7" ... <entr+"</entr+" ... </ro7"
      </t0od+"
    </tgroup"
  </informaltable"
```

```
...
</sect1>
```

TestLink uses such element as data for just one requirement. This element is defined via constant `DOCBOOK_REQUIREMENT` (check the code). i.e. `<sect3>` is default but could be modified.

Each requirement content is maintain the following way:

title – receive text in tag `<title>`

req_doc_id – parse title for the first two words and add counter. You can modify regular expression directly in code. Default is `"[a-zA-Z_]*[0-9]*"`.

description – parse following elements after title (`<para>`, `<orderedlist>`, `<informaltable>`, etc.). DocBook elements are modified to HTML tags. Unknown ones are omitted.

*Warning: the original code could be modified to fit your structure of DocBook. Check `requirement.inc.php` (**function** `importReqDataFromDocBook($fileName)`) and related constants.*

Warning: generated `REQ_DOC_ID` is danger for the case of update. Because it's generated from file content without relation to existing testlink data.

2.8 Results import

Results import is supported from TL 1.7.

Example 1: Results in XML format (using internal ID)

```
<?xml version="1.0" encoding="UTF-8" ?>
<results>
  <testcase id="100" <$ ?'D internal/'2 id "
    <result"p</result"
    <notes"functionalit+ works great </notes"
  </testcase>
  <testcase id="@00" "
    <result"f</result"
    <notes"t-is case failed due to error </notes"
  </testcase>
  <testcase id="1;0" "
    <result"0</result"
    <notes"t-is test case is blocked</notes"
  </testcase>
</results>
```

Example 2: Results in XML format (using external ID)²

```
<?xml version="1.0" encoding="UTF-8" ?>
<$ &comment "
<results>
  <testcase external_id="413 1" "
    <$ if not present logged #ser 7ill 0e #sed "
    <tester"#011I</tester" <$ tester 31.?E Eame "
    <$ if not present no7JK 7ill 0e #sed "
    <timestamp"@00! 0= 0! 1:D00D00</timestamp"
    <result"p</result"
    <notes"functionalit+ works great </notes"
  </testcase>

  <testcase external_id="413 1" " <$ (E1T,6/ 696 for *(56 test case "
    <result"f</result"
    <notes"functionalit+ works great 8?5?</notes"
  </testcase>
```

2 Format supported on TL 1.8 beta 3 and UP

```

<testcase external_id="1@;G" "    <$    Using ?ET6/E(3 ?'    "
    <res#lt"f</res#lt"
    <notes"Using ?ET6/E(3 ?' as link </notes"
</testcase"
</res#lts"

```

You can import several / multiple execution results using a single XML file

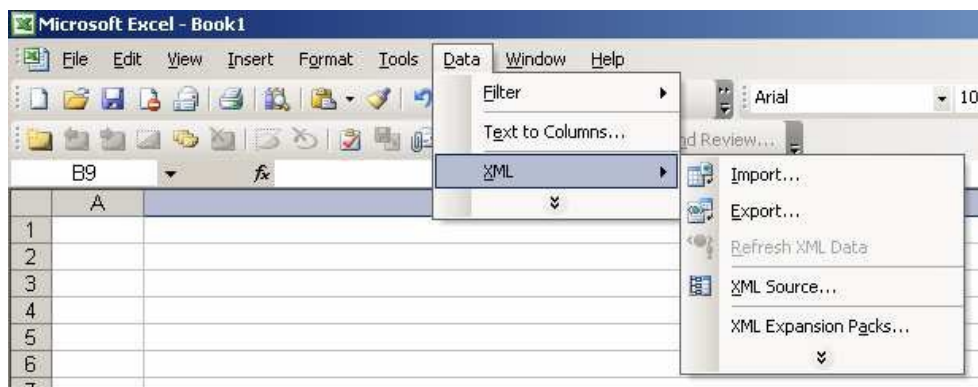
2.9 Import Test Cases from Excel via XML

Creating XML file to import in TestLink

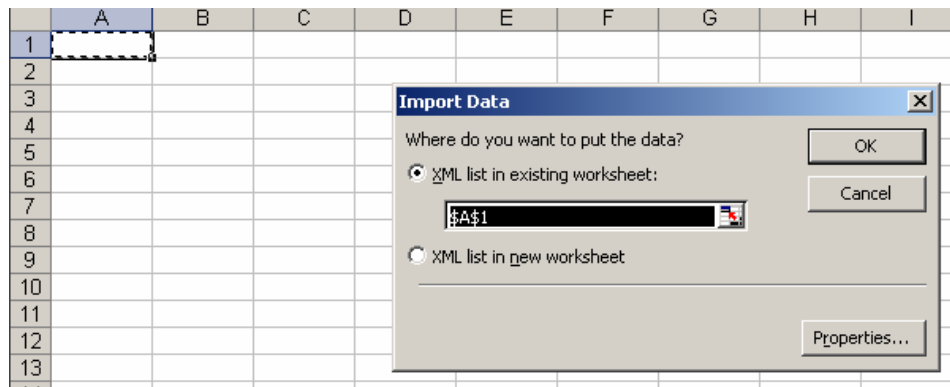
Step 1. Export one or more dummy Test Cases from TestLink into a XML file.

Step 2. Open new blank spread sheet document file.

Step 3. Navigate through menu bar Data > XML > Import & select the sample XML file. It creates appropriate structure in Excel.



Step 4. Then we will get dialog box asking "Where do you want to put the data?"



Step 5. Choose option one "XML list in existing worksheet" with first cell \$A\$1

Step 6. You will be able to see following columns : name, summary, steps & expected results

	A	B	C	D
1	name	summary	steps	expectedresults
			user has logged in the application.	On navigating to 'Accounts' screen, the following view should be displayed.
			Navigate to 'Accounts' screen by clicking on the Accounts screen	'My Accounts List' view (By default this view should be displayed).
2	ACC 1.1	Whether the user can view A tab.		'All Accounts List' view.
				be loaded.
			i) Click on the 'Accounts' screen.	It should display all the following applets:
3	ACC 1.2	Whether the required applet	ii) Check for applets	
4	*			
5				

Step 7. Copy your data into this file accordingly & save the file in XML Data (*.xml) format

Step 8. Check your XML file for correctness by opening with the help of internet explorer.

```

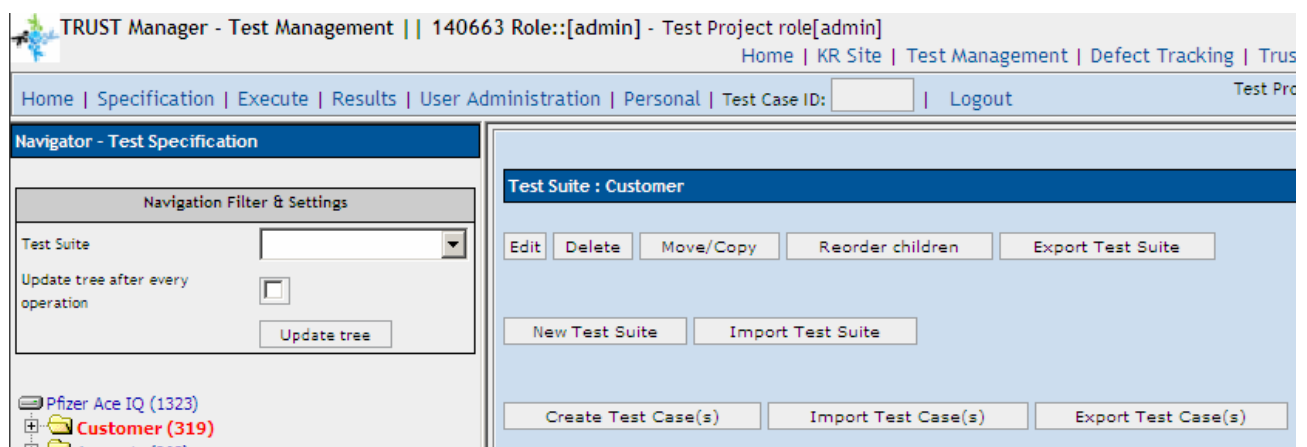
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
- <testcases>
- <testcase name="ACC 1.1">
  <summary>Whether the user can view Accounts List View as
    a default view under Accounts screen.</summary>
  <steps>Pre-condition: A valid user has logged in the
    application. Navigate to 'Accounts' screen by clicking on
    the Accounts screen tab.</steps>
  <expectedresults>On navigating to 'Accounts' screen, the
    following view should be displayed. 'My Accounts List'
    view (By default this view should be displayed). 'All
    Accounts List' view.</expectedresults>
</testcase>
- <testcase name="ACC 1.2">
  <summary>Whether the user can view Accounts List View as
    a default view under Accounts screen.</summary>
  <steps>Pre-condition: A valid user has logged in the
    application. Navigate to 'Accounts' screen by clicking on
    the Accounts screen tab.</steps>
  <expectedresults>On navigating to 'Accounts' screen, the
    following view should be displayed. 'My Accounts List'
    view (By default this view should be displayed). 'All
    Accounts List' view.</expectedresults>
</testcase>
</testcases>

```

Importing XML file into TestLink

Step 1. Login in to TestLink > Select your project in dropdown list.

Step 2. Click on Specification > Create New Suite > Select Suite > Click on Import Test Cases



Step 3. Browse for the XML file, submit it and you are done with the Importing.

2.10 Platforms

Platforms can be both imported and exported. The feature is available from TL 1.9

Platform Management	
Platform	Description
MAC OS	
Solaris 10	New Solaris
Solaris 8	
Solaris 9	
Windows 2008	
Windows 7	New MS OS
<input type="button" value="Create platform"/> <input type="button" value="Export"/> <input type="button" value="Import"/>	

Example of data:

```
<?xml version="1.0" encoding="UTF-8"?>
<platforms>
  <platform>
    <name><T(5 1)></name>
    <notes><T( )></notes>
  </platform>
  <platform>
    <name><T(*olaris 10)></name>
    <notes><T(e7 *olaris)></notes>
  </platform>
  <platform>
    <name><T(*olaris !)></name>
    <notes><T( )></notes>
  </platform>
  <platform>
    <name><T(*olaris =)></name>
    <notes><T( )></notes>
  </platform>
  <platform>
    <name><T(Mindo7s @00!)></name>
    <notes><T( )></notes>
  </platform>
  <platform>
    <name><T(Mindo7s N)></name>
    <notes><T(e7 5* 1)></notes>
  </platform>
</platforms>
```

2.11 Custom Fields

Definition of custom field can be both in imported and exported since TL 1.9.

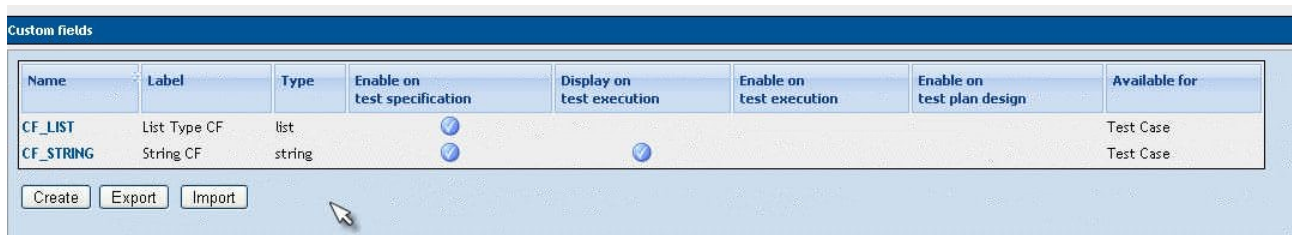


Illustration 2: Custom field management window with Import and Export actions

Example of definition file:

```
<?xml version=L1.0L encoding=L?*1 !!:= 1L?"
<c#stom_fields"
  <c#stom_field"
    <name"<$%&' (T(%&F_*T/?E.)) "</name"
    <la0el"<$%&' (T(%*tring &F)) "</la0el"
    <t+pe"<$%&' (T(%0)) "</t+pe"
    <possible_val#es"<$%&' (T(%)) "</possible_val#es"
    <defa#lt_val#e"<$%&' (T(%)) "</defa#lt_val#e"
    <valid_regexp"<$%&' (T(%)) "</valid_regexp"
    <lengt-_min"<$%&' (T(%0)) "</lengt-_min"
    <lengt-_max"<$%&' (T(%0)) "</lengt-_max"
    <s-o7_on_design"<$%&' (T(%1)) "</s-o7_on_design"
    <ena0le_on_design"<$%&' (T(%1)) "</ena0le_on_design"
    <s-o7_on_exec#tion"<$%&' (T(%1)) "</s-o7_on_exec#tion"
    <ena0le_on_exec#tion"<$%&' (T(%0)) "</ena0le_on_exec#tion"
    <s-o7_on_testplan_design"<$%&' (T(%0)) "</s-o7_on_testplan_design"
    <ena0le_on_testplan_design"<$%&' (T(%0)) "</ena0le_on_testplan_design"
    <node_t+pe_id"<$%&' (T(%I)) "</node_t+pe_id"
  </c#stom_field"
  <c#stom_field"
    <name"<$%&' (T(%&F_3?*T)) "</name"
    <la0el"<$%&' (T(%3ist T+pe &F)) "</la0el"
    <t+pe"<$%&' (T(%G)) "</t+pe"
    <possible_val#es"<$%&' (T(%'eep 4#rpleO<esOF#een)) "</possible_val#es"
    <defa#lt_val#e"<$%&' (T(%)) "</defa#lt_val#e"
    <valid_regexp"<$%&' (T(%)) "</valid_regexp"
    <lengt-_min"<$%&' (T(%0)) "</lengt-_min"
    <lengt-_max"<$%&' (T(%0)) "</lengt-_max"
    <s-o7_on_design"<$%&' (T(%1)) "</s-o7_on_design"
    <ena0le_on_design"<$%&' (T(%1)) "</ena0le_on_design"
    <s-o7_on_exec#tion"<$%&' (T(%0)) "</s-o7_on_exec#tion"
    <ena0le_on_exec#tion"<$%&' (T(%0)) "</ena0le_on_exec#tion"
    <s-o7_on_testplan_design"<$%&' (T(%0)) "</s-o7_on_testplan_design"
    <ena0le_on_testplan_design"<$%&' (T(%0)) "</ena0le_on_testplan_design"
    <node_t+pe_id"<$%&' (T(%I)) "</node_t+pe_id"
  </c#stom_field"
</c#stom_fields"
```