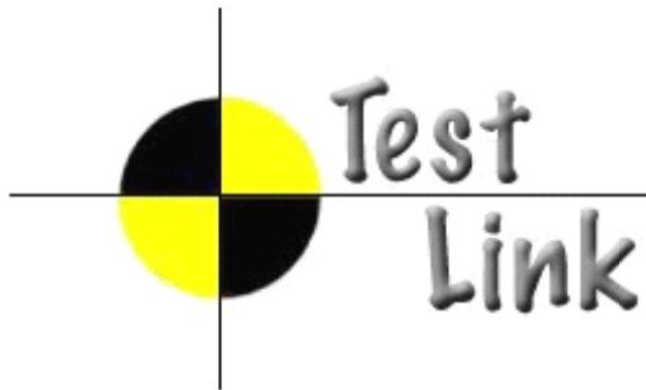


Testlink Convert V6.15.7.x 基础教程

(2015/7/2)

原创作者: xmyemail@foxmail.com



QQ 交流群: 188370086 (Testlink Convert 最新版本、文档分析, 提供 Testlink 技术交流)

微信公众号: testlinkconvert (关注微信, 获赠免费激活码, 具体参考微信活动详情)

淘宝网店: <http://shop62067219.taobao.com>

PS: 本工具不以盈利为目的, 仅象征性收取维护成本, 维护不易, 请尊重知识产权。

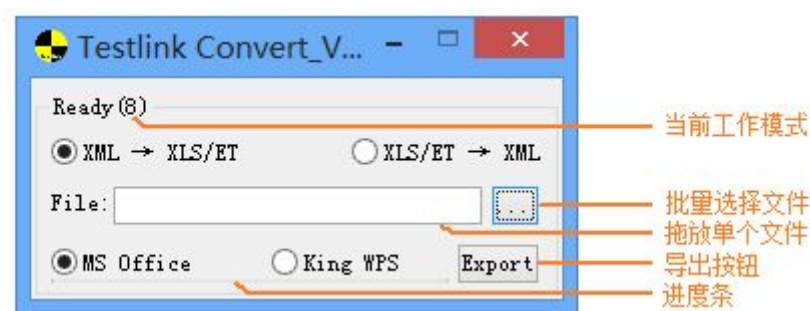
目录

一、简介.....	3
1、概要.....	3
2、版本变更记录.....	4
3、为什么要使用 Testlink Convert?	4
4、Testlink Convert 可以导出/导入哪些信息?	4
5、不同的转换模式（Mode）有什么区别?	5
二、安装/激活.....	5
三、XML/XLS 格式转换.....	6
3.1、Testlink V1.9.x.....	6
3.1.1、Mode=1.....	6
3.1.2、Mode=6.....	8
3.1.3、Mode=7.....	10
3.1.4、Mode=8.....	10
3.1.5、Mode=9.....	11
3.2、Testlink V1.8.x.....	12
3.2.1、Mode=18.....	12
3.2.2、Mode=98.....	13
3.3、REQ 需求规约模式（ Requirement Specification）.....	13
3.3.1、Mode=51.....	13
3.3.2、Mode=52.....	14
3.3.3、Mode=53.....	14
3.4、批量转换文件.....	15
四、高级应用（Testlink.ini）.....	16
4.1、参数列表.....	17
4.2、工具目录下没有 Testlink.ini 文件，如何生成?	18
4.3、如何制作带有自定义节点且符合自身需求的用例模版?	19
4.4、如何根据 Excel 定义 Testlink.ini 中的 Custom Field 参数?	20
4.5、开启 Excel 文件截取功能.....	21
4.6、启用 XML 转 XLS 时，如何对步骤或结果进行智能编号?	22
4.7、如何自定义 Excel 中的列属性?	23

一、简介

1、概要

Testlink Convert 是 Testlink 用例维护的第三方辅助工具，提供 XML、Excel 文件之间一键式相互转换，该工具支持多种转换模式，满足不同层次用户的需要，完美实现“Test suite、Case ID、Title、Summary、Preconditions、Step actions、Expected Results、Execution、Execution type、Test importance、Keywords、Requirements、Custom Fields”工具 13 个节点信息导出/导入，弥补了 Testlink 系统自身的不足，该工具功能更强大灵活，操作简单，效率极高，是 Testlink 用例维护的最佳工具。



工具特性：

- 1、兼容 Testlink V1.8.x 与 Testlink V1.9.x 系统。
- 2、支持 Testcase、Testsuite 方式导入/导出，且完美实现子步骤、子结果分步式导入/导出。
- 3、支持需求导入（工作模式：51/52/53）。
- 4、支持多种用例模版样式（1/18/6/7/8/98），满足不同 Excel 书写格式需求。
- 4、支持 Excel 任意行与行之间的数据或任意多工作表中的表数据截取转化。
- 5、智能匹配用例等级、用例执行方式，实现数字与字符之间智能转化。
- 6、通过配置参数更改，可获得额外个性定制功能。

系统要求： Windows XP /Vista/WIN7/WIN8/Windows Server

安装要求： 必须安装 Microsoft Office 或 King WPS 至少一款 Office 软件。（建议使用 WPS）

软件作者： xmyemail@foxmail.com

注意事项：

- 1、部分杀软（如，360）可能会误报病毒，请方式放心使用，该应用程序绿色无毒。
- 2、该工具不定期更新，如需获取新版本，请关注 BLOG 更新信息。

2、版本变更记录

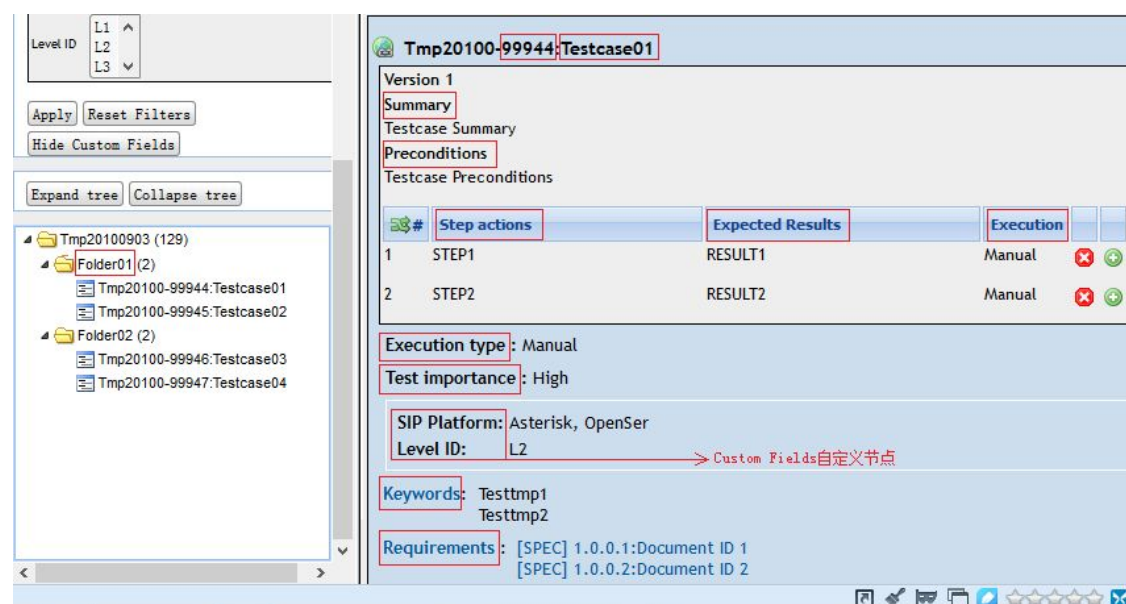
版本	修改信息
6.15.7.1	标题中多个空格转码后无法正常导入 Testlink 的 Bug。
	MODE=6 模式下，未拼合的文件列会识别为 Testcase，已修改为无文件夹名称识别为 Testcase，有文件夹名称识别为文件夹。
	增加 KeepStyle 配置接口，用于自定义是否将 XLS 中的空白转换为 XML 能识别的代码，从而保持排版样式，默认 1。
	去除 Automark 参数，替换为 SetStepID、SetResultID 参数，增加 StepSTR_inResult 参数。
	XML 转 XLS 时，替换为新的 StepID、ResultID 计算方法，遇到空 Step、Result 时，自动跳过。
	解决 XML 转 XLS 时，Summary、Step、Result 中的空格被自动去除的问题。

3、为什么要使用 Testlink Convert?

- 1>Testlink 导入/导出功能不完善，无法满足 Excel 导入/导出需求。
- 2>单靠人力上传用例费时费力，工作效率低下。
- 3>想在 Testlink 中集成 Excel 导入/导出功能，但又不精通 PHP 代码。
- 4>常常维护用例，重新编辑已有用例的过程很繁琐。

4、Testlink Convert 可以导出/导入哪些信息？

工具可完美实现“Test suite、Case ID、Title、Summary、Preconditions、Step actions、Expected Results、Execution type、Test importance、Keywords、Requirements、Custom Fields”工具 13 个节点信息导出/导入，请参考如下示意图，红色部分均能导出/导入。



5、不同的转换模式（Mode）有什么区别？

为兼顾不同格式需求，工具支持多种 Excel 用例模版，方便适配公司对用例书写格式要求，不同的模版区别在于采用不同的方式解析 Excel，具体情况如下：

用例导出/导入：

Mode	Testlink 版本	Excel 多工作表	目录树	分步式导出/导入
1	V1.9.x	√	√	×
6	V1.9.x	√	√，采用单元格拼合识别目录树	√
7	V1.9.x	√	√，采用不同工作表识别目录树	×
8	V1.9.x	√	√，采用不同工作表识别目录树	√
9	V1.9.x	√	√，采用单元格拼合识别目录树	×
98	V1.8.x	√	√，采用单元格拼合识别目录树	×

Ps: Mode=81 为 Mode=8 的扩展样式，具体参考 Mode=8 演示效果。

需求规约导入：（仅支持 Excel 转 XML）

Mode	Testlink 版本	目录树
5	V1.9.x	×
51	V1.9.x	√，采用单元格拼合识别目录树
52	V1.9.x	√，采用不同工作表识别目录树

二、安装/激活

注意：

- 1、执行安装前将杀毒软件暂停或将工具加入信任列表，否则可能无法完成软件激活或运行。
- 2、工具依赖MircoSoft Office或King WPS，因此必须安装其中一款Office软件。

激活步骤：

- 1、下载最新的 Testlink Convert 工具，解压。
- 2、运行"Testlink Convert"主程序，获得随机码。
- 3、将随机码发送至阿里旺旺“伴偶梦一场”或发送 Email 至 xmyemail@foxmail.com。
- 4、收到激活码，输入对应的激活码即可。

注：激活后的 PC 升级到新版本时无需再次激活。

三、XML/XLS 格式转换

3.1、Testlink V1.9.x

3.1.1、Mode=1

参考模板：Example(1)_Testcase_Default.xls

Excel 转 XML：

1、Mode=1 即 Testcase 模式，支持 Excel、XML 互转，由工具转换出的 XML 不支持目录树和分步式导入/导出，XML 导入 Testlink 时，应选择“Import testcase”方式导入，适合用户维护。书写格式如下：

1>工作表：用例可以写在一个或多个工作表中（工具会自动将每个工作表独立转换出一个 XML 文件）。

2>关键字/规约编号：多个关键或多个规约编号必须以“,”号隔开。

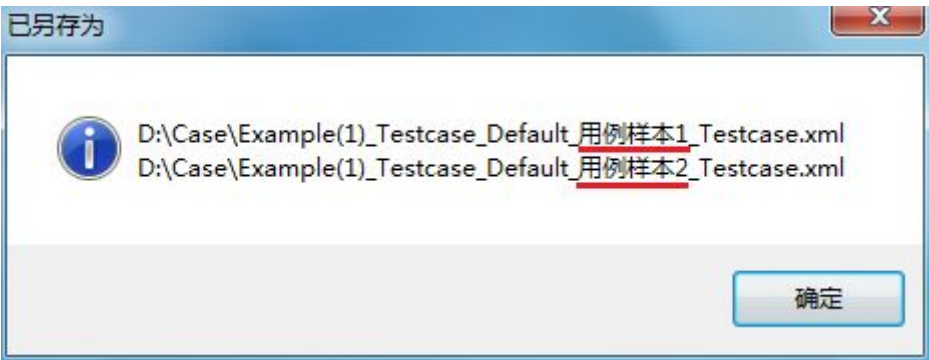
3>用例等级：数字（Testlink 默认只识别 1-3）或英文单词 High、Medium、Low。

4>执行方式：数字（Testlink 默认只识别 1-2）或英文单词 Manual、Automated。

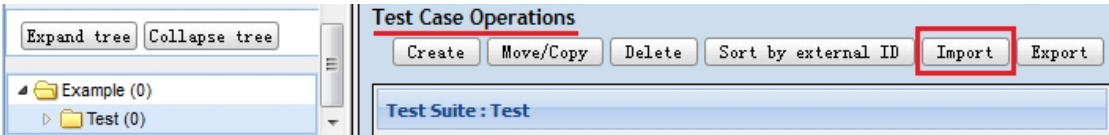
5>操作步骤/期望结果：无论是多个步骤或多个结果，均各自集成在各自的单元格中。

用例名称	用例等级	执行方式	关键字	用例摘要	预置条件	操作步骤	预期结果	测试方式	规约编号
用例1	High	Automated	API, GUI	用例1摘要	用例1条件	1、用例1步骤1 2、用例1步骤2	1、用例1结果		1.0.0.1, 1.0.0.2
用例2	3	2	WEB			1、用例2步骤	1、用例2结果		1.0.0.3
用例3			API, GUI	用例3摘要	用例3条件	1、用例3步骤	1、用例1结果		
用例4	3	Automated				1、用例4步骤	1、用例4结果		1.0.0.3

2、由于该 Excel 存在多个工作表，因此，工具会自动将每个工作表独立转换出一个 XML 文件，转换效果如下：



3、将上图所示 XLS 文件转换为 XML 文件，进入 Testlink 中的一个项目，选择一个文件夹，然后使用 Testcase Import 方式导入。

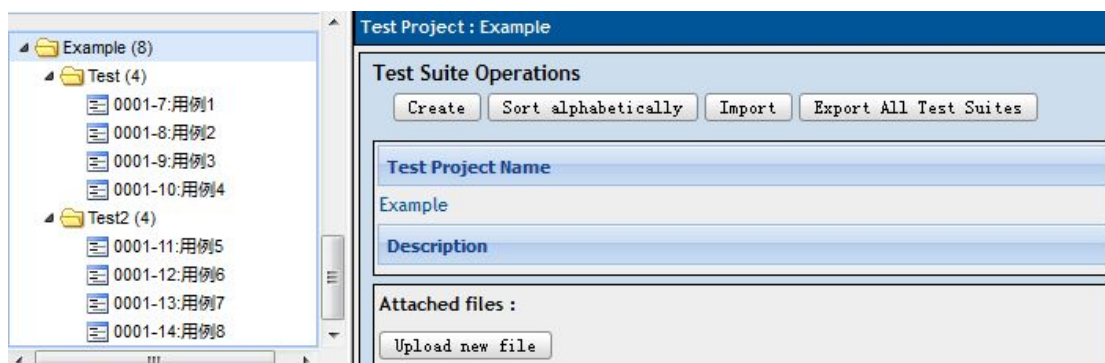


4、将“Example(1)_Testcase_Default_用例样本 1_Testcase.xml”导入到 Testlink，效果如下：



XML 转 Excel:

1、该模式下，工具不识别 XML 中的目录树，仅会提取 XML 中的用例，并保存在一个工作表中，参考如下：



(Example 项目下的用例)

2、将 Example 项目导出为 XML 文件，工具 XML 转为 Excel 效果如下：

CaseName	Importance	Execution	Keywords	Summary	Preconditions	StepActions	ExpectedResults	ExecutionT	RequirementID
用例1	3	2	API, GUI	用例1摘要	用例1条件	1、用例1步骤1 2、用例1步骤2	1、用例1结果		1.0.0.1,1.0.0.2
用例2	3	2	WEB	用例2摘要	用例2条件	1、用例2步骤1 2、用例2步骤2	1、用例2结果		
用例3	2	1	API, GUI	用例3摘要	用例3条件	1、用例3步骤1 2、用例3步骤2	1、用例3结果		
用例4	3	2		用例4摘要	用例4条件	1、用例4步骤1 2、用例4步骤2	1、用例4结果		
用例5	3	2	API, GUI	用例5摘要	用例5条件	1、用例5步骤1 2、用例5步骤2	1、用例5结果		1.0.0.1,1.0.0.2
用例6	3	2	WEB	用例6摘要	用例6条件	1、用例6步骤1 2、用例6步骤2	1、用例6结果		
用例7	2	1	API, GUI	用例7摘要	用例7条件	1、用例7步骤1 2、用例7步骤2	1、用例7结果		
用例8	3	2		用例8摘要	用例8条件	1、用例8步骤1 2、用例8步骤2	1、用例8结果		

(Example 项目转为 Excel 效果)

3.1.2、Mode=6

模版文件：Example(6)_Testsuite_Default.xls

Excel 转 XML:

1、该模式为 Testsuite 模式，适合大多数公司用例模版，支持目录树、多工作表以及分步式导入/导出，该模式下，XLS 转换出的 XML 导入 Testlink 时，应选择“Import testsuit”方式导入。书写格式：

1>工作表：用例可以写在一个或多个工作表中（工具会自动将每个工作表独立转换出一个 XML 文件）。

2>关键字/规约编号：多个关键或多个规约编号必须以“,”号隔开。

3>用例等级：数字（Testlink 默认只识别 1-3）或英文单词 High、Medium、Low。

4>执行方式：数字（Testlink 默认只识别 1-2）或英文单词 Manual、Automated。

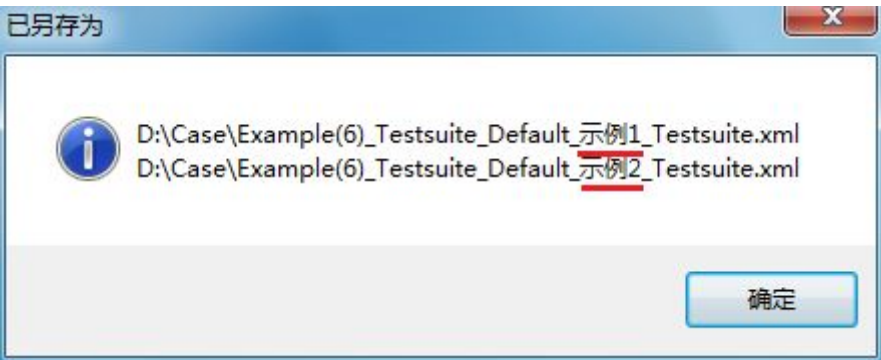
5>操作步骤/期望结果：多个步骤或结果可以写在多个单元格中，步骤与结果一一对应。

6>模块：即目录树名称，必须拼合，如图所示，“标题 1、标题 2”属于“文件夹 1”目录下的用例。

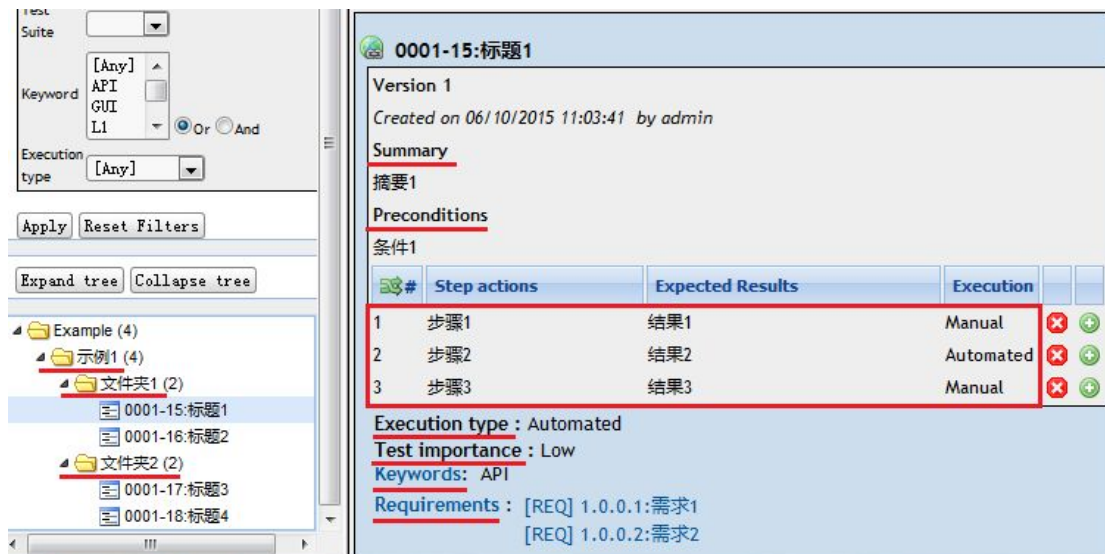
4>单元格样式：一个用例有多个步骤或结果，可以如果所示，拼合 B-G、K 列（参考蓝色框），使用例更具可阅读性，此外，可以不必拼合（参考绿色框）。

A	B	C	D	E	F	G	H	I	J	K
模块	标题	用例等级	执行方式	关键字	用例摘要	预置条件	操作步骤	预期结果	测试方式	规约编号
文件夹1	标题1	1	2	API	摘要1	条件1	步骤1	结果1	1	1.0.0.1, 1.0.0.2
							步骤2	结果2	2	
							步骤3	结果3	1	
文件夹2	标题2	2	1	GUI, API	摘要2	条件2	2步骤1	2结果1	1	1.0.0.2
							2步骤2	2结果2	1	
	标题3	1	2	API	摘要3	条件3	3步骤1	3结果1	1	1.0.0.1
							3步骤2	3结果2	2	
							3步骤3	3结果3	1	
	标题4	2	1	GUI, API	摘要4	条件4	4步骤1	4结果1	1	1.0.0.2
							4步骤2	4结果2		

2、将以上 Excel 转为 XML，工具自动将多个工作表独立存为一个 XML：



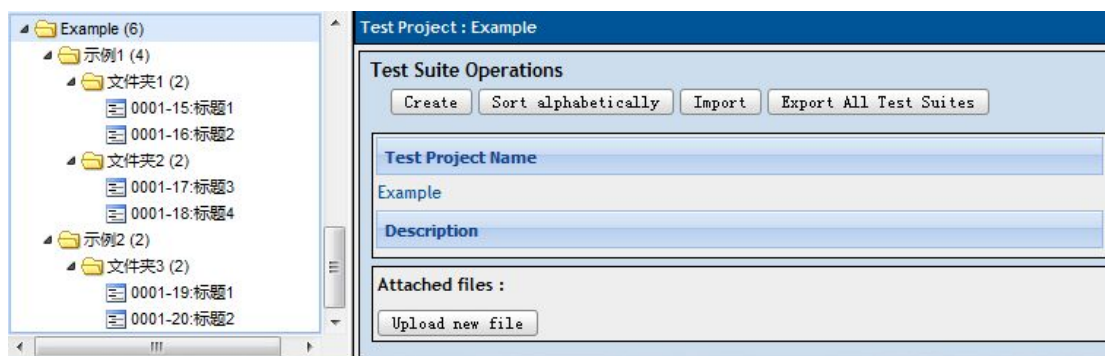
3、将“Example(6)_Testsuite_Default_示例 1_Testsuite.xml”通过 Testsuite Import 方式导入 Testlink，如下图所示，工具会将多个步骤和结果进行对应，分步式导入到 Testlink 中，效果如下：



(转换后的效果，支持分步式导入)

XML 转 Excel:

1、工具在执行 XML 转 Excel，仅会对最后一层目录树进行处理，如下图，将 Example 项目导出 XML 文件:



2、通过工具，将 XML 转为 Excel 效果如下: (仅处理最后一层目录树，参考“文件夹3”)

FolderName	CaseName	Importance	Execution	Keywords	Summary	Preconditions	StepActions	ExpectedResults	ExecutionT	RequirementID
文件夹1	标题1	1		2 API	摘要1	条件1	步骤1	结果1	1	1.0.0.1, 1.0.0.2
	标题2	2		1 API, GUI	摘要2	条件2	步骤2	结果2	2	
	标题3	1		2 API	摘要3	条件3	步骤3	结果3	1	
文件夹2	标题4	2		1 API, GUI	摘要4	条件4	2步骤1	2结果1	1	1.0.0.2
	标题1	1		2 API	摘要1	条件1	2步骤2	2结果2	1	
	标题2	2		1 API, GUI	摘要2	条件2	3步骤1	3结果1	1	1.0.0.1
文件夹3	标题1	1		2 API	摘要1	条件1	3步骤2	3结果2	2	
	标题2	2		1 API, GUI	摘要2	条件2	3步骤3	3结果3	1	
	标题3	1		2 API	摘要3	条件3	4步骤1	4结果1	1	1.0.0.2
	标题4	2		1 API, GUI	摘要4	条件4	4步骤2	4结果2	1	
	标题1	1		2 API	摘要1	条件1	步骤1	结果1	1	1.0.0.1
	标题2	2		1 API, GUI	摘要2	条件2	步骤2	结果2	2	
	标题3	1		2 API	摘要3	条件3	步骤3	结果3	1	
	标题4	2		1 API, GUI	摘要4	条件4	2步骤1	2结果1	1	1.0.0.2
	标题5	1		2 API	摘要5	条件5	2步骤2	2结果2	1	

(“示例2”被丢弃, “文件夹3”被提取)

3.1.3、Mode=7

该模式为 Testsuite 模式，适合维护多个目录下的用例，支持目录树，该模式下，XLS 转换出的 XML 导入 Testlink 时应选择“Import testsuit”方式导入，与此同时，XLS 文件应该按照以下格式书写：（如图所示）

- 1>Excel 中的“工作表”即表示一个“文件夹”，该文件夹下的用例即该工作表中的所有数据。
- 2>多个关键或多个规约编号必须以“,”号隔开。
- 3>用例等级可以是数字或英文单词 High、Medium、Low。（自定义字符串请参考 Testlink.ini 高级应用）
- 4>执行方式可以是数字或英文单词 Manual、Automated。（自定义字符串请参考 Testlink.ini 高级应用）
- 5>无论是多个步骤或多个结果，均各自集成在各自的单元格中。

PS：与 Mode=1 相比，这个模式多支持了文件夹样式。

A	B	C	D	E	F	G	H	I	J
用例名称	用例等级	执行方式	关键字	用例摘要	预置条件	操作步骤	预期结果	测试方式	规约编号
用例1	High	Automated	API, GUI	用例1摘要	用例1条件	1、用例1步骤1 2、用例1步骤2	1、用例1结果		1.0.0.1, 1.0.0.2
用例2	3	2	WEB	多个关键字已头号隔开		1、用例2步骤	1、用例2结果		1.0.0.3

3.1.4、Mode=8

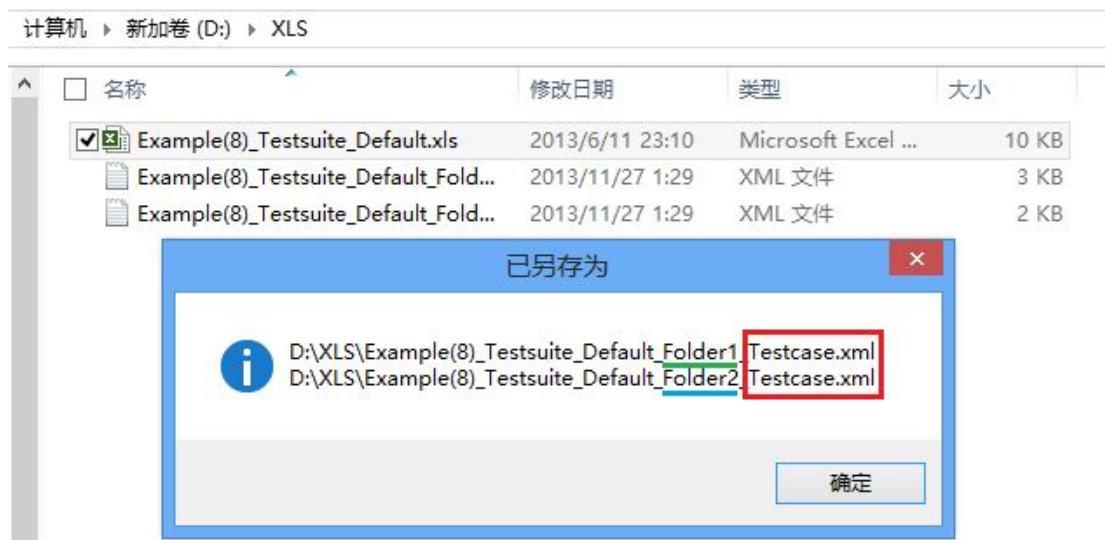
用例模版：Example(8&81)_Testsuite_Default.xls

该模式为 Testsuite 模式，支持目录树与子步骤子结果方式导出/导入，该模式下，XLS 转换出的 XML 导入 Testlink 时应选择“Import testsuit”方式导入，与此同时，XLS 文件应该按照以下格式书写：（如图所示）

- 1>工作表：Excel 中的每个“工作表”均即表示一个“文件夹”，该文件夹下的用例即该工作表中的所有数据。
- 2>关键字/规约编号：多个关键或多个规约编号必须以“,”号隔开。
- 3>用例等级：数字或英文单词 High、Medium、Low。
- 4>执行方式：数字或英文单词 Manual、Automated。
- 5>操作步骤/预期结果：一个步骤对应一个结果和“测试方式”，如果下图所示，A 至 F 列可以是拼合单元格，也可以不拼合。

A	B	C	D	E	F	G	H	I	J
用例名称	用例等级	执行方式	关键字	用例摘要	预置条件	操作步骤	预期结果	测试方式	规约编号
用例格式1	High	Automated	API, GUI	用例1摘要	用例1条件	用例1步骤1 用例1步骤2	步骤1结果 步骤2结果	Manual Automated	1.0.0.1, 1.0.0.2
用例格式2	3	2	WEB			用例2步骤1 用例2步骤2	步骤1结果 步骤2结果	1 2	1.0.0.3
用例格式3	3	Automated	WEB			用例3步骤	用例3结果		1.0.0.3

如上图所示，如果想将 Folder1、Folder2 分别拆分并转换成独立的 XML，此时，可以将 Mode 参数设置为 81，通过工具转换效果如下，值得注意的是，拆分后输出的 XML 为 Testcase 模式，参考红色标记处。



3.1.5、Mode=9

该模式为 Testsuite 模式，适合维护多个目录下的用例，支持目录树，但不支持子步骤子结果方式导出/导入，该模式下，XLS 转换出的 XML 导入 Testlink 时应选择“Import testsuit”方式导入（多工作表除外），与此同时，XLS 文件应该按照以下格式书写：（如图所示）

- 1>所有用例均写在第一个工作表中。（程序仅读取第一个工作表中的数据）
- 2>多个关键或多个规约编号必须以“,”号隔开。
- 3>用例等级可以是数字或英文单词 High、Medium、Low。（自定义字符串请参考 Testlink.ini 高级应用）
- 4>执行方式可以是数字或英文单词 Manual、Automated。（自定义字符串请参考 Testlink.ini 高级应用）
- 5>无论是多个步骤或多个结果，均各自集成在各自的单元格中。
- 6>拼合的单元格表示文件夹，从该拼合单元开始到下一个拼合单元之前的数据属于该文件夹中的用例。

Example(9&98)_...Default.xls X									
A	B	C	D	E	F	G	H	I	J
用例名称	用例等级	执行方式	关键字	用例摘要	预置条件	操作步骤	预期结果	测试方式	规约编号
文件夹1									
用例1	High	Automated	API, GUI	用例1摘要	用例1条件	1、用例1步骤1 2、用例1步骤2	1、用例1结果		1.0.0.1, 1.0.0.2
用例2	3	2	WEB			1、用例2步骤	1、用例2结果		1.0.0.3
文件夹2									
用例3			API, GUI	用例3摘要	用例3条件	1、用例3步骤	1、用例1结果		
用例4	3	Automated				1、用例4步骤	1、用例4结果		1.0.0.3

如果在 Mode=9 模式下存在多个工作表，那么工具会自动拆分每个工作表，并判断每个工作表中是否有存在文件夹样式，若存在文件夹则以 Testsuit 方式导出，若不存在文件夹，则以 Testcase 方式导出 XML。如下图所示，XLS 存在两个工作表：

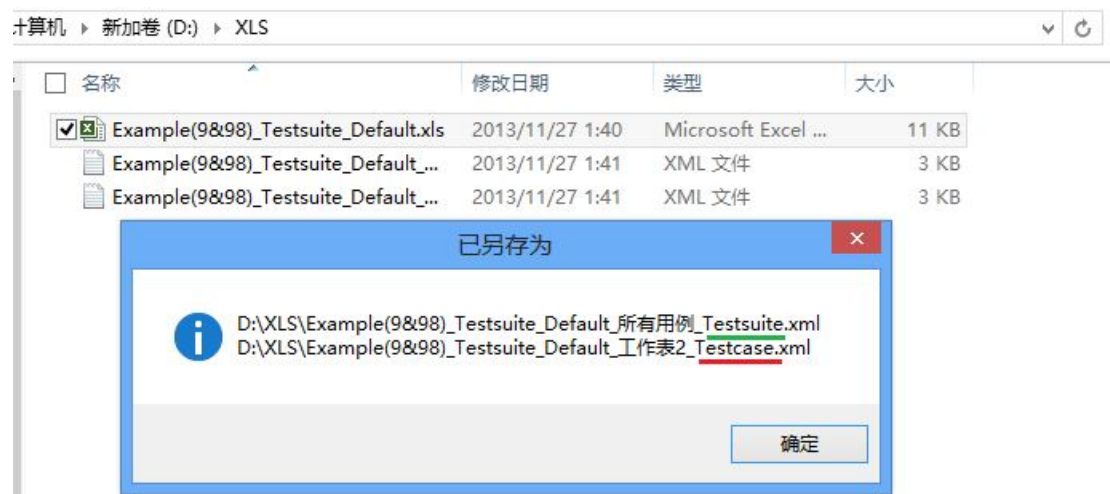
Worksheet1

用例名称	用例等级	执行方式	关键字	用例摘要	预置条件	操作步骤	预期结果	测试方式	规约编号
用例1	High	Automated	API, GUI	用例1摘要	用例1条件	1、用例1步骤1 2、用例1步骤2	1、用例1结果		1.0.0.1, 1.0.0.2
用例2	3	2	WEB			1、用例2步骤	1、用例2结果		1.0.0.3

Worksheet2

用例名称	用例等级	执行方式	关键字	用例摘要	预置条件	操作步骤	预期结果	测试方式	规约编号
用例3			API, GUI	用例3摘要	用例3条件	1、用例3步骤	1、用例1结果		
用例4	3	Automated				1、用例4步骤	1、用例4结果		1.0.0.3

通过工具抓换后的效果：



3.2、Testlink V1.8.x

3.2.1、Mode=18

略，参考 Mode=1 对应的 XLS 模板。

3.2.2、Mode=98

略，参考 Mode=9 对应的 XLS 模板。

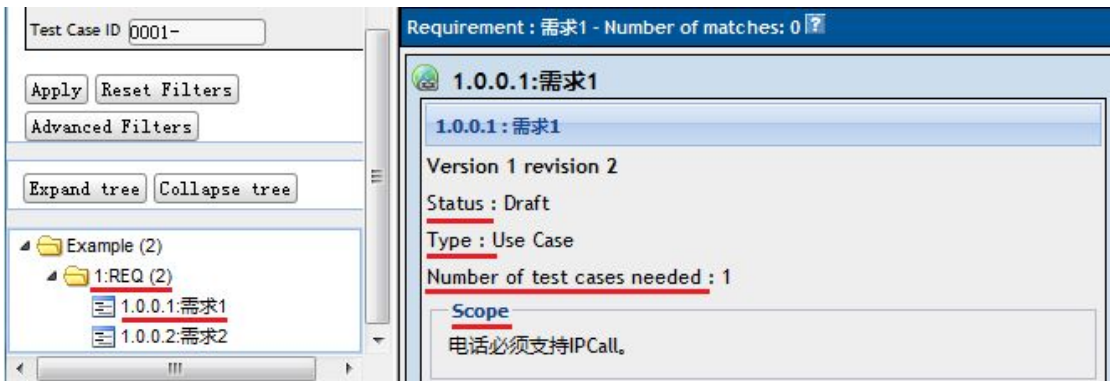
3.3、REQ 需求规约模式（ Requirement Specification）

工具仅支持需求规约 XLS 转 XML 格式，具体可使用 Mode=51/52/53 三种 XLS 样式，支持 8 个节点的信息转换，样式参考模版参考：

Example(51)_REQ_Default.xls，不支持目录数结构。

Example(52)_REQ_Suite_Default.xls，支持一个级别的目录树结构。

Example(53)_REQ_Suite_Default.xls，支持一个级别的目录树结构。



（工具支持的节点，参考红色标记处）

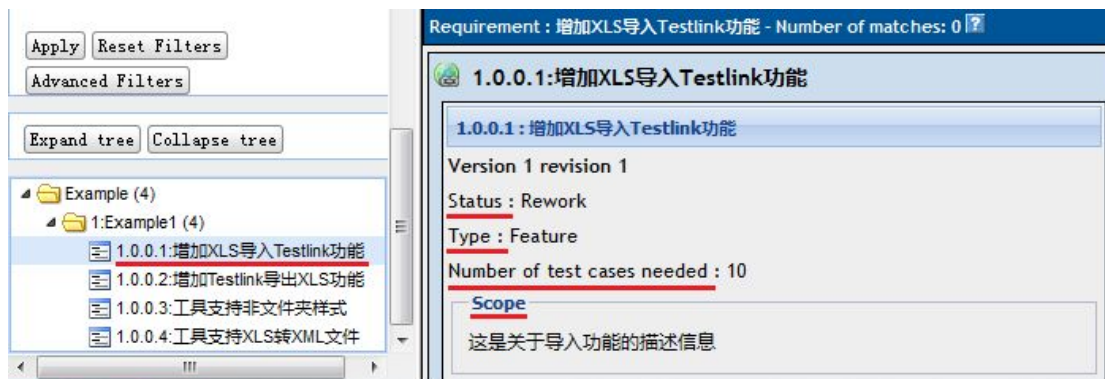
3.3.1、Mode=51

参考样式模版：Example(51)_REQ_Default.xls，该模式不支持目录树，所有 REQ 需求规约都在同一个 XLS 工作表中，工具转换出的 XML 应使用“Requirement Operations”方式导入 Testlink。参考如下效果：

J23						
	A	B	C	D	E	F
1	REQ ID	REQ Title	Description	Status	Type	Number
2	1.0.0.1	增加XLS导入Testlink功能	这是关于导入功能的描述信息	3	2	10
3	1.0.0.2	增加Testlink导出XLS功能	这是关于导出功能的描述信息	Review	Feature	
4	1.0.0.3	工具支持非文件夹样式	这是关于非文件夹功能的描述	1	2	8
5	1.0.0.4	工具支持XLS转XML文件	这是关于XLS转XML功能的描述	Finish	Feature	5
6						
7						

（REQ XLS 样本）

导入到 Testlink 后的显示效果：



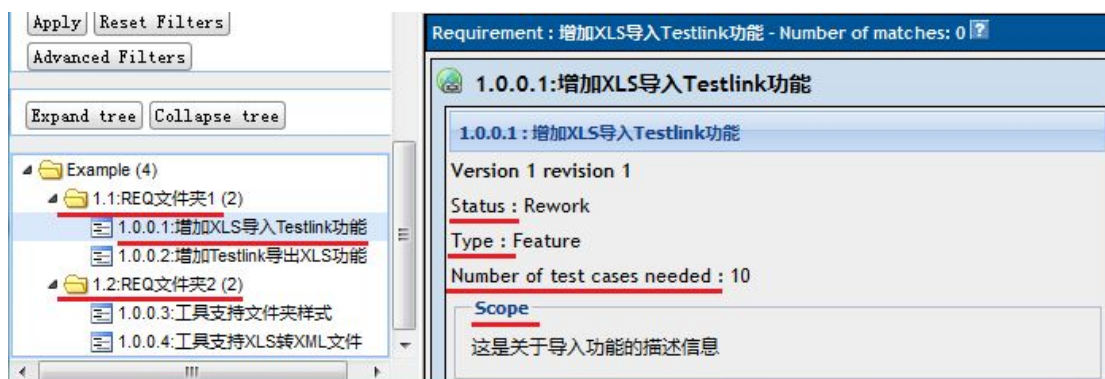
3.3.2、Mode=52

参考样式模版：Example(52)_REQ_Suite_Default.xls，该样式支持一个级别的目录树，导入 Testlink 时，应选择“Requirement Specification Operations”方式导入，书写格式如下：

- 1>文件夹应采用拼合单元格来表示。
- 2>所有 REQ 需求规约均写在 XLS 中的同一个工作表中。

REQ ID	REQ Title	Description	Status	Type	Number
1.1	REQ文件夹1				
1.0.0.1	增加XLS导入Testlink功能	这是关于导入功能的描述信息	3	2	10
1.0.0.2	增加Testlink导出XLS功能	这是关于导出功能的描述信息	Review	Feature	
1.2	REQ文件夹2				
1.0.0.3	工具支持文件夹样式	这是关于文件夹功能的描述	1	2	8
1.0.0.4	工具支持XLS转XML文件	这是关于XLS转XML功能的描述	Finish	Feature	5

导入到 Testlink 后的显示效果：



3.3.3、Mode=53

参考样式模版：Example(53)_REQ_Suite_Default.xls，该样式支持一个级别的目录树，导入 Testlink 时，应选择“Requirement Specification Operations”方式导入，书写格式如下：

- 1>文件夹和文件夹 ID 应采用工作表名称表示，以“|”间隔，例如“1.1|用户需求”。
- 2>一个工作表中仅包含该目录下的所有需求。

	A	B	C	D	E	
	REQ ID	REQ Title	Description	Status	Type	Num
1	1.0.0.1	增加XLS导入Testlink功能	这是关于导入功能的描述信息	3	2	
2	1.0.0.2	增加Testlink导出XLS功能	这是关于导出功能的描述信息	Review	Feature	
3						
4						
5						
6						
7						
8						

导入到 Testlink 后的显示效果:

Requirement : 增加XLS导入Testlink功能 - Number of matches: 0

1.0.0.1:增加XLS导入Testlink功能

1.0.0.1: 增加XLS导入Testlink功能

Version 1 revision 1

Status : Rework

Type : Feature

Number of test cases needed : 10

Scope

这是关于导入功能的描述信息

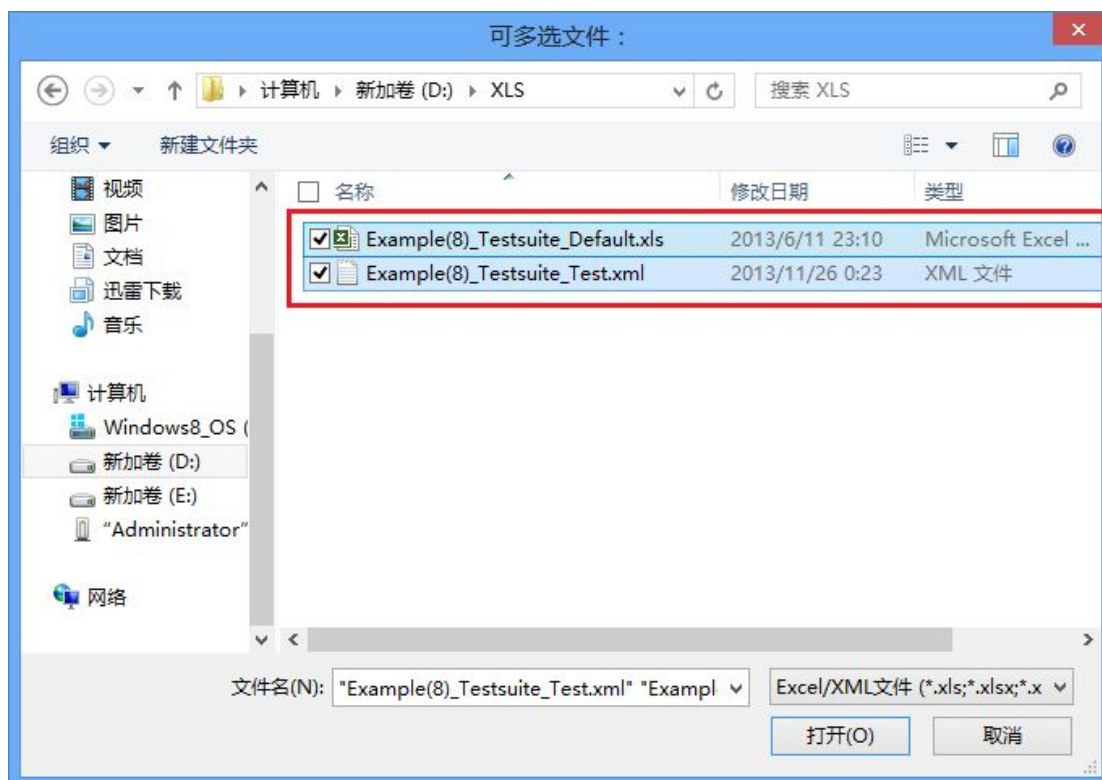
3.4、批量转换文件

工具支持 XML 与 XLS 文件混合多选，工具会根据相应的格式自行判断，然后转换出对应的文件。使用此模式时遵循以下规则：

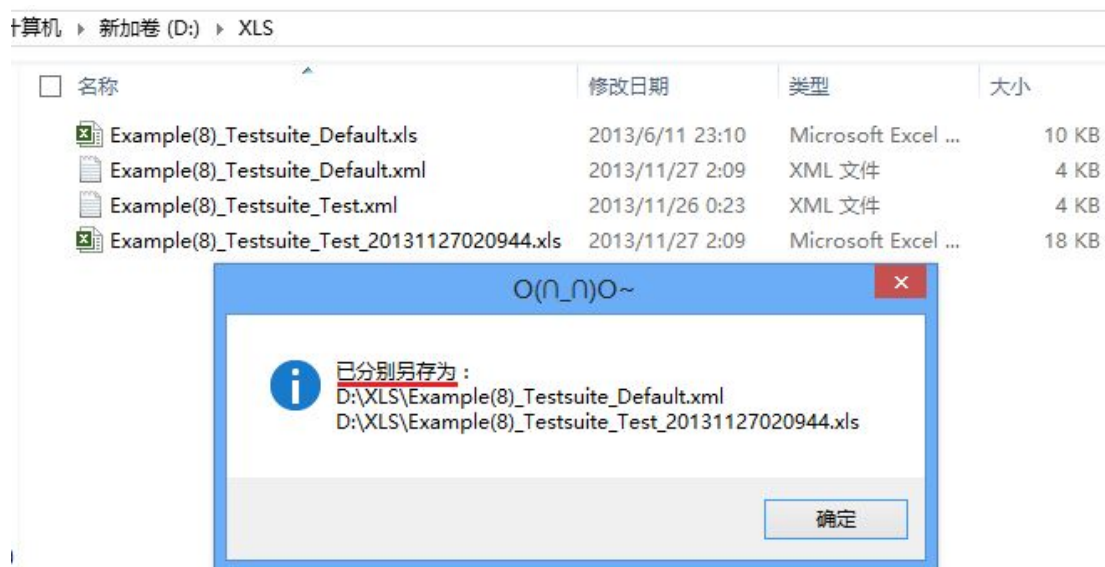
- 1>多文件必须可同时选择 XML 和 XLS 文件，所选择的 XML/XLS 文件必须属于同一个 Mode 模式。
- 2>仅支持同个文件夹下多选文件。

操作步骤：

- 1、打开工具后，点击工具界面上的浏览按钮打开“多选文件”对话框。如下图，可同时选择 XML/XLS。



2、选定后点击打开按钮确认，此时工具自动判断格式并转换。如下图：



四、高级应用（Testlink.ini）

针对一些高级用户，允许通过配置文件 Testlink.ini 自定义个性化功能，包括 XLS 标题定制、XLS 读取列、开启在 XLS 中支持自定义项（Custom Field）、Excel 截取、等级与执行方式识别以及不同模式切换。

4.1、参数列表

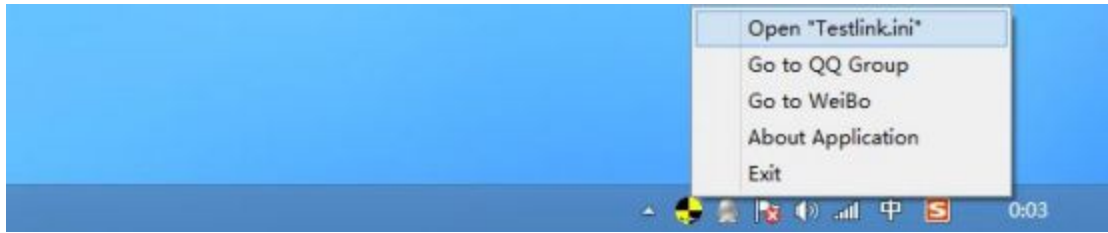
字段	参数	备注
MOD	Mode	定义转换模式，默认为 8，支持 1/18/51/52/53/7/8/9/98 模式。
	GUIDefOption	定义 GUI 窗口中 XLS/XML 转换的缺省项。 1，GUI 界面默认选中“XML -> XLS/ET”选项。 2，GUI 界面默认选中“XLS/ET -> XML”选项。
	GUIDefOffice	定义 GUI 缺省使用 Microsoft Office 或 King WPS。 1，GUI 界面默认选中“MS Office”选项。 2，GUI 界面默认选中“King WPS”选项。
CFG	Defexecution	定义用例缺省执行方式，XLS 中的 Execution 为空时使用该值。
	Defimportance	定义用例缺省等级，XLS 中的 Importance 为空时使用该值。
	Strdisplay	XML 转 XLS 时，将 XML 中的 Execution、Importance 值替换成字符串，需要与 Executionstr、Importancestr 配合使用。 0，不做处理，按照 XML 中的值进行显示。 1，替换为字符串。
	Executionstr	定义 XML 中 Execution 值对应的字符串。
	Importancestr	定义 XML 中 Importance 值对应的字符串。
XLS	RowDetection	0，自动判断 XLS 中的行数。 1，使用 Excel 截取模式截取指定的数据。
	EnableCustomField	0，关闭 CustomField 模式。 1，打开对 CustomField 节点的转换支持。
	IDTag	定义 XIDColumn 参数所指列的第一行文本显示。
	TitleTag	定义 XTitleColumn 参数所指列的第一行文本显示。
	ImportanceTag	定义 XImportanceColumn 参数所指列的第一行文本显示。
	ExecutionTag	定义 XExecutionColumn 参数所指列的第一行文本显示。
	KeywordTag	定义 XKeywordColumn 参数所指列的第一行文本显示。
	SummaryTag	定义 XSummaryColumn 参数所指列的第一行文本显示。
	ConditionTag	定义 XConditionColumn 参数所指列的第一行文本显示。
	StepTag	定义 XStepActionColumn 参数所指列的第一行文本显示。
	ResultTag	定义 XResultColumn 参数所指列的第一行文本显示。
	ExecutionTypeTag	定义 XExecutionTypeColumn 参数所指列的第一行文本显示。
	RequirementIDTag	定义 XRequirementIDColumn 参数所指列的第一行文本显示。
	CustomFieldTag	该文本作为自定义字段的<Name>，具有实际意义，必须与 Testlink 中 Custom Fields 对应，多个字段时采用范围格式，例如，API GUI，则说明有两个自定义字段，分别为<API>、<GUI>。
	XIDColumn	可选，定义 Case ID 所在的列，0 表示忽略 ID 转换。
	XTitleColumn	必选，定义 Title 所在的列，默认为 1，不能为 0。
	XImportanceColumn	必选，定义 Importance 所在的列，默认为 2，不能为 0。
	XExecutionColumn	必选，定义 Execution 所在的列，默认为 3，不能为 0。
	XKeywordColumn	必选，定义 Keyword 所在的列，默认为 4，不能为 0。
	XSummaryColumn	必选，定义 Summary 所在的列，默认为 5，不能为 0。
	XConditionColumn	必选，定义 Condition 所在的列，默认为 6，不能为 0。

	XStepActionColumn	必选，定义 StepAction 所在的列，默认为 7，不能为 0。
	XResultColumn	必选，定义 Result 所在的列，默认为 8，不能为 0。
	XExecutionTypeColumn	必选，定义 ExecutionType 所在的列，默认为 9，不能为 0。
	XRequirementIDColumn	必选，定义 RequirementID 所在的列，默认为 10，0 表示忽略。
	XCustomFieldColumn	必选，定义 CustomField 所在的列，默认为 11，不能为 0。
XML	SetStepID	XML->XLS 时，对步骤进行自动编号。 0，不智能编号。 1，启用智能编号。
	SetResultID	XML->XLS 时，对结果进行自动编号。 0，不智能编号。 1，启用智能编号。
	Punctuation	SetStepID、SetResultID 功能开启后，定义自动编号时使用的标点符号。
	StepSTR_inResult	智能在结果中标识出对应的步骤，例如， #1 ，表示步骤 1，该参数为空时，不加入步骤标识。
	TagCRLF	全局变量，请不要更改此参数，否则程序无法正常运行。
	SuiteStrTrans	仅在 Mode=7/8/53 中生效，用于自动判断是否存在相同目录树，请不要更改此参数，否则程序无法正常运行。
REQ	StatusDEF	定义缺省的 Status 。
	TypeDEF	定义缺省的 Type 。
	NumberDEF	定义缺省的 Number 。
	StatusSTR	与 Testlink 中 Status 显示的文本保持一致。
	StatusRES	StatusSTR 文本对应的解析字符。
	TypeSTR	与 Testlink 中 Type 显示的文本保持一致。
	XREQ_docid	定义 Document ID 所在的列，默认 1，表示 A 列。
	XREQ_doctitle	定义 Document Title 所在的列，默认 2，表示 B 列。
	XREQ_Scope	定义 Document Scope 所在的列，默认 3，表示 C 列。
	XREQ_Status	定义 Document Status 所在的列，默认 4，表示 D 列。
	XREQ_Type	定义 Document Type 所在的列，默认 5，表示 F 列。
	XREQ_Number	定义 Number of testcases need 所在的列，默认 6，表示 G 列。
	StringRegExpTitle	默认空，用于批量正则删除 Title 中的字符。
	StringRegExpScope	默认空，用于批量正则删除 Scope 中的字符。

4.2、工具目录下没有 Testlink.ini 文件，如何生成？

可按照以下方法开启该功能：

- 1、运行 **Testlink Convert** 工具。
- 2、从工具的托盘菜单中打开“**Testlink.ini**”配置文件。
- 3、此时工具目录下就会生成一个默认的配置文。



开启 Custom Field 功能

与 Custom Field 相关的配置参数共有三个，分别为 EnableCustomField、CustomFieldTag、XCustomFieldColumn，是决定能否正确转换的关键因素，具体含义参考列表中的解释。

4.3、如何制作带有自定义节点且符合自身需求的用例模版？

Testlink Convert 提供 Custom Field 自定义关键字扩展功能，该功能默认处于关闭状态，使用该功能前，需要先启用 Custom Field 支持，相关联的参数如下：

EnableCustomField=1

；0，不启用 CustomField；1，启用 CustomField。

CustomFieldTag=Server|Level

；自定义关键字名称，多个关键字时使用“|”间隔，不建议使用带有特殊符号的关键字。

XCustomFieldColumn=11-12

；自定义关键字在 XLS 中所处的列编号，例如，11 列对应 Server，12 列对应 Level。

在不清楚如何在 Excel 中定义 Custom Field 格式时，可将现有 Testlink 中带有 Custom Field 的 XML 导出来，然后再通过工具转换出一个标准的 Excel 模板（转换过程集中工具会自动配置相关参数），按照该模板书写就可以反导入 Testlink，已确保 100%兼容 Custom Field。下面演示制作符合自身需求的用例模版案例：

例如，有如下一个 Testcase，该用例中包含 Server、Level 两个自定义关键字，首先我们从 Testlink 中导出这个 Testcase 为 XML 文件，将该 XML 用例转为 XLS 后效果如下：

Test Case Summary:

- Summary Text
- Preconditions
- Precondition Text

#	Step actions	Expected Results	Execution
1	Actions Test1	Result Text1	Manual
2	Actions Test2	Result Text2	Automated

Execution type : Automated
Test importance : High

Server: Linux, WIN8
Level: L4

Keywords: API, GUI

Requirements : [这是测试标题1] 2.0.0.1:增加XLS导入Testlink功能

Excel Export (Test_20131...0400.xls):

CaseName	Importance	Execution	Keywords	Summary	Preconditions	StepActions	ExpectedResults	ExecutionType	RequirementID	Server	Level
Case1	3	2	API, GUI	Summary Text	Precondition Text	Actions Test1	Result Text1	1	2.0.0.1	Linux WIN8	L4
						Actions Test2	Result Text2	2			

Ps: 如果自定义关键字属于多选属性, 则在 XLS 中以“|”间隔对应的多个值, 例如, Testlink 中显示 Server 的值是 Linux,WIN8, 则在 XLS 中表示为“Linux|WIN8”。

将 XML 文件拖入工具后, 工具自动检查 XML 中的用户自定义节点, 会发现与配置文件不匹配, 此时, 点击“是”让工具帮助你自动根据当前的 XML 配置 CustomFieldTag、XCustomFieldColumn 参数并转化, 然后会导出一个 Excel 文件 (参考上图的导出结果), 而这个 XLS 文件可以作为用例模版, 方便你可将新写的 XLS 用例导入到 Testlink, 而且完美支持自定义节点。



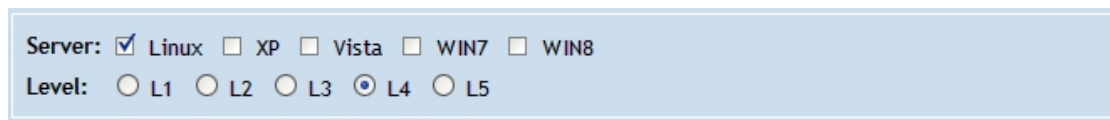
此时, 你可以重新打开 Testlink.ini, 工具根据 XML 自动配置参数如下:

```
CustomFieldTag=Server|Level
XCustomFieldColumn=11-12
```

该参数表示 Excel 中的第 11 列对应的自定义 Name 是 Server, 第 12 列对应的自定义 Name 是 Level。当然, 如果你熟悉配置这些配置规则, 就可以自行定义, 而无需使用 XML 转出的模板。

4.4、如何根据 Excel 定义 Testlink.ini 中的 Custom Field 参数?

1、首先确定你的 Testlink 用例中有多少个关键字，如下，有 Server、Level 两个自定义节点。



2、打开 Testlink.ini 文件，找到“EnableCustomField”，并将值修改为 1，开启 Custom Field 功能。

3、根据#1 步配置 CustomFieldTag、XCustomFieldColumn 参数。

CustomFieldTag=Server|Level =>表示本次转换的 Excel 中有两个自定义节点。多个自定义节点用“|”间隔。

XCustomFieldColumn=11-12 =>表示 Excel 中的第 11 列（即 K 列）为 Server，第 12 列（即 L 列）为 Level，那么工具在读取 Excel 时，会将 11-12 列作为 Custom Field 的节点，并转为 XML。该参数仅支持区间表示法，例如，有 5 个节点，即 11-15。

4.5、开启 Excel 文件截取功能

Testlink Convert 强大之处不仅是简单的实现 Excel、XML 文件转换，考虑到用户实际使用场景，工具加入了对 Excel 进行自定义数据截取功能，其中包括工作表之间的截取、行与行之间截取，非常灵活，该功能使用适合用例维护时使用。

截取模式作为工具的扩展功能，默认未开启该功能，因此需要使用该功能的用户必须自行在配置文件“Testlink.ini”中设定参数，可按照以下方法开启该功能：

1、找到配置文件 Testlink.ini 中的“RowDetection”字段，并将值设定为 1，保存即可。

RowDetection=1

2、将一个 Excel 用例拖入到工具中，此时工具弹出如下提示框。

3、输入截取的命令即可截取指定数据并转换为 XML 文件。



执行截取的基本命令格式为“\$Sheet:\$Row1-\$Row2”，可使用英文逗号“,”隔开多个命令进行扩展，

例如：

1、1,3,5,7 表示截取 Excel 文件中的第 1、3、5、7 个工作表并转换为 XML 文件。

2、1,3:10-30,5 表示截取 Excel 文件中的第 1、5，以及第 3 个工作表中的第 10 至 30 行之间的数据，并将这些转换为 XML 文件。

3、3:10-30,5:20-40 表示截取 Excel 文件中的第 3 个工作表中第 10 至 30 行之间的数据，以及第 5 个工作表中第 20 至 40 行之间的数据，并将这些转换为 XML 文件。

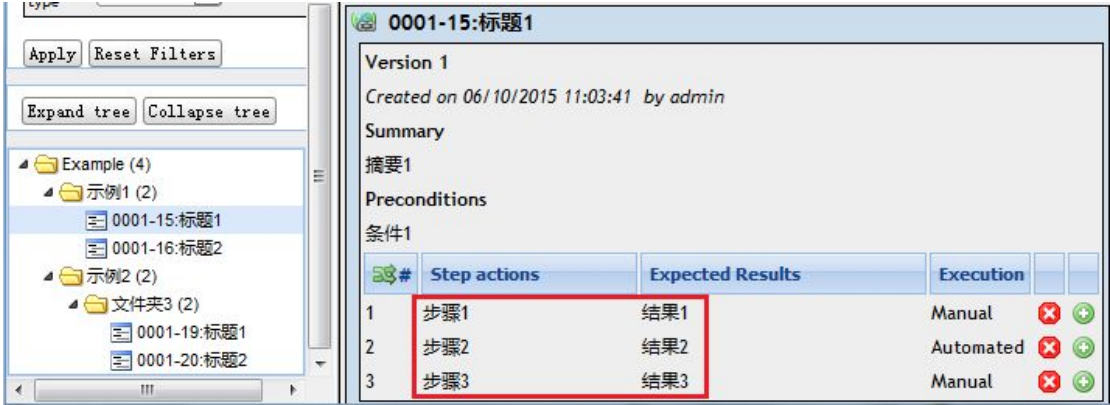
PS:

1>工具版本必须大于等于 V6.7.13.0905，低于这个版本的童鞋请更新版本。

2>该功能仅支持 Excel 转 XML 执行截取。

4.6、启用 XML 转 XLS 时，如何对步骤或结果进行智能编号？

至 Testlink V1.9.x 开始，创建用例步骤或结果时，系统会自动生成步骤编号，工具 XML 转换为 XLS 时，支持智能编号计算，若需对 XML 导出 XLS 的用例步骤、结果增加编号，可按照以下步骤执行：



1、打开 Testlink.ini，启用 SetStepID、SetResultID，保存参数。

SetStepID=1

；步骤编号：0，不智能编号；1，启用智能编号。

SetResultID=1

；结果编号：0，不智能编号；1，启用智能编号。

Punctuation=、

；编号所使用的标点符号，即 ID+“、”样式。

StepSTR_inResult=#

；智能在结果中标识出对应的步骤，例如，#1，表示步骤 1，该参数为空时，不加入步骤标识。

2、按照上面的参数配置，工具 XML 转 XLS 时将会自动对步骤和结果编号，以“标题 1”用例为例，转换效果如下：

CaseName	Importance	ExecutionKeywords	Summary	Precondition	StepActions	ExpectedResults	ExecutionRequirementID
标题1	1	2 API	摘要1	条件1	1、步骤1 2、步骤2 3、步骤3	1、#1, 结果1 2、#2, 结果2 3、#3, 结果3	1 1.0.0.1, 1.0.0.2 2 1

Ps: 例如，“1、#1 结果 1”，其中“、”即 Punctuation 参数，“#1”中的“#”即 StepSTR_inResult 参数。

3、参数配置为 SetStepID=1、SetResultID=0 时演示效果如下：（仅对步骤进行智能编号）

CaseName	Importance	ExecutionKeywords	Summary	Precondition	StepActions	ExpectedResults	ExecutionRequirementID
标题1	1	2 API	摘要1	条件1	1、步骤1 2、步骤2 3、步骤3	结果1 结果2 结果3	1 1.0.0.1, 1.0.0.2 2 1

4、参数配置为 SetStepID=1、SetResultID=1、StepSTR_inResult 为空时，演示效果如下：（步骤和结果均编号，但结果中不包含步骤对应关系）

CaseName	Importance	ExecutionKeywords	Summary	Precondition	StepAction	ExpectedResult	ExecutionType	RequirementID
标题1	1	2 API	摘要1	条件1	1、步骤1 2、步骤2 3、步骤3	1、结果1 2、结果2 3、结果3	1 2 1	1.0.0.1, 1.0.0.2

4.7、如何自定义 Excel 中的列属性？

工具通过识别 Excel 中的列顺序进行判断属性，因此，用户可以根据自己的需求，调整不同的属性列数序，例如 A-J 列：

A	B	C	D	E	F	G	H	I	J
用例名称	用例等级	执行方式	关键字	用例摘要	预置条件	操作步骤	预期结果	测试方式	规约编号
用例格式1	High	Automated	API, GUI	用例1摘要	用例1条件	用例1步骤1 用例1步骤2	步骤1结果 步骤2结果	Manual Automated	1.0.0.1, 1.0.0.2
用例格式2	3	2	WEB			用例2步骤1 用例2步骤2	步骤1结果 步骤2结果	1 2	1.0.0.3
用例格式3	3	Automated	WEB			用例3步骤	用例3结果		1.0.0.3

可通过 Testlink.ini 自定义识别的列：

```

XFolderColumn=0
;=>$XFolderColumn,Only support Mode=6
XIDColumn=0
XTitleColumn=1 ;=>用例标题对应 Excel 中的 A 列。
XImportanceColumn=2 ;=>用例等级对应 Excel 中的 B 列。
XExecutionColumn=3 ;=>执行方式对应 Excel 中的 C 列。
XKeywordColumn=4 ;=>关键字对应 Excel 中的 D 列。
XSummaryColumn=5 ;=>用例摘要对应 Excel 中的 E 列。
XConditionColumn=6 ;=>用例条件对应 Excel 中的 F 列。
XStepActionColumn=7 ;=>用例步骤对应 Excel 中的 G 列。
XResultColumn=8 ;=>用例结果对应 Excel 中的 H 列。
XExecutionTypeColumn=9 ;=>步骤执行方式对应 Excel 中的 I 列。
XRequirementIDColumn=10 ;=>用例标题对应 Excel 中的 J 列。
XCustomFieldColumn=11 ;=>用例标题对应 Excel 中的 K 列。

```

例如，将用例 Excel 属性列定义依次定义为用例标题、用例摘要、预置条件、操作步骤、预期结果、用例等级、执行方式、关键字、测试方式、规约编号，如下 Excel：

A	B	C	D	E	F	G	H	I	J
用例名称	用例摘要	预置条件	操作步骤	预期结果	用例等级	执行方式	关键字	测试方式	规约编号
用例格式1	用例1摘要	用例1条件	用例1步骤1 用例1步骤2	步骤1结果 步骤2结果	High	Automated	API, GUI	Manual Automated	1.0.0.1, 1.0.0.2
用例格式2			用例2步骤1 用例2步骤2	步骤1结果 步骤2结果	3	2	WEB	1 2	1.0.0.3
用例格式3			用例3步骤	用例3结果	3	Automated	WEB		1.0.0.3

那么需要工具适配新的 XLS 模版，则需要通过 Testlink.ini 重定义属性列：

XFolderColumn=0
;=>\$XFolderColumn,Only support Mode=6
XIDColumn=0
XTitleColumn=1 ;=>用例标题对应 Excel 中的 A 列。
XImportanceColumn=6 ;=>用例等级对应 Excel 中的 F 列。
XExecutionColumn=7 ;=>执行方式对应 Excel 中的 G 列。
XKeywordColumn=8 ;=>关键字对应 Excel 中的 H 列。
XSummaryColumn=2 ;=>用例摘要对应 Excel 中的 B 列。
XConditionColumn=3 ;=>用例条件对应 Excel 中的 C 列。
XStepActionColumn=4 ;=>用例步骤对应 Excel 中的 D 列。
XResultColumn=5 ;=>用例结果对应 Excel 中的 E 列。
XExecutionTypeColumn=9 ;=>步骤执行方式对应 Excel 中的 I 列。
XRequirementIDColumn=10 ;=>用例标题对应 Excel 中的 J 列。
XCustomFieldColumn=11 ;=>用例标题对应 Excel 中的 K 列。