

目录

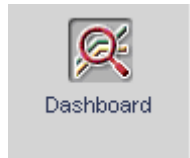
1. 引言	2
2. 创建 excel 报告	2
3. 编写 sql 查询语句	2
a) 导出全部的 sql 语句	3
b) 按测试集导出 sql 语句	4
c) 按测试树主题目录 sql 语句	5
4. Ecxcel 执行代码	6
5. Excel 设置	8
6. Generate 生成文档	8

1. 引言

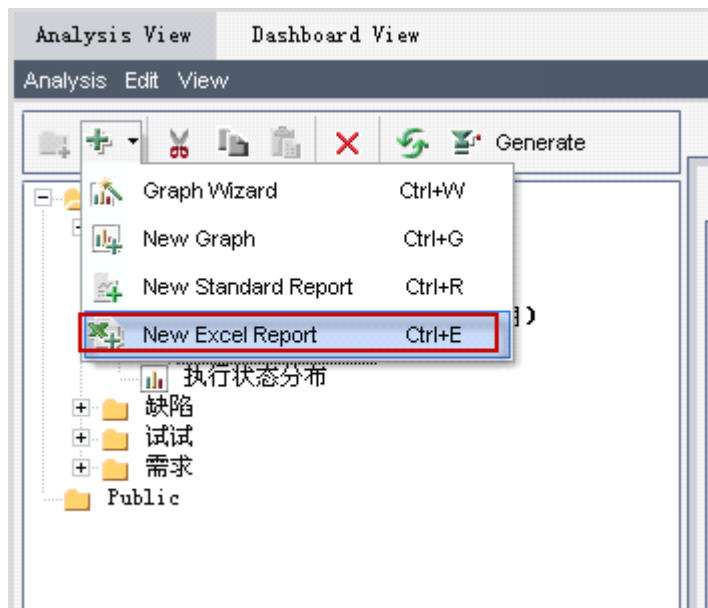
QC 工具本身没有提供导出含测试步骤的测试用例。这个对我们来说就很不方便，我们在导出测试用例时，总是希望能导出含测试步骤的测试用例。后面我进行了研究，现在把这个方法总结分享给大家，具体如下介绍。其中本文介绍的有关操作主要基于 QC 的 10 版本。

2. 创建 excel 报告

在

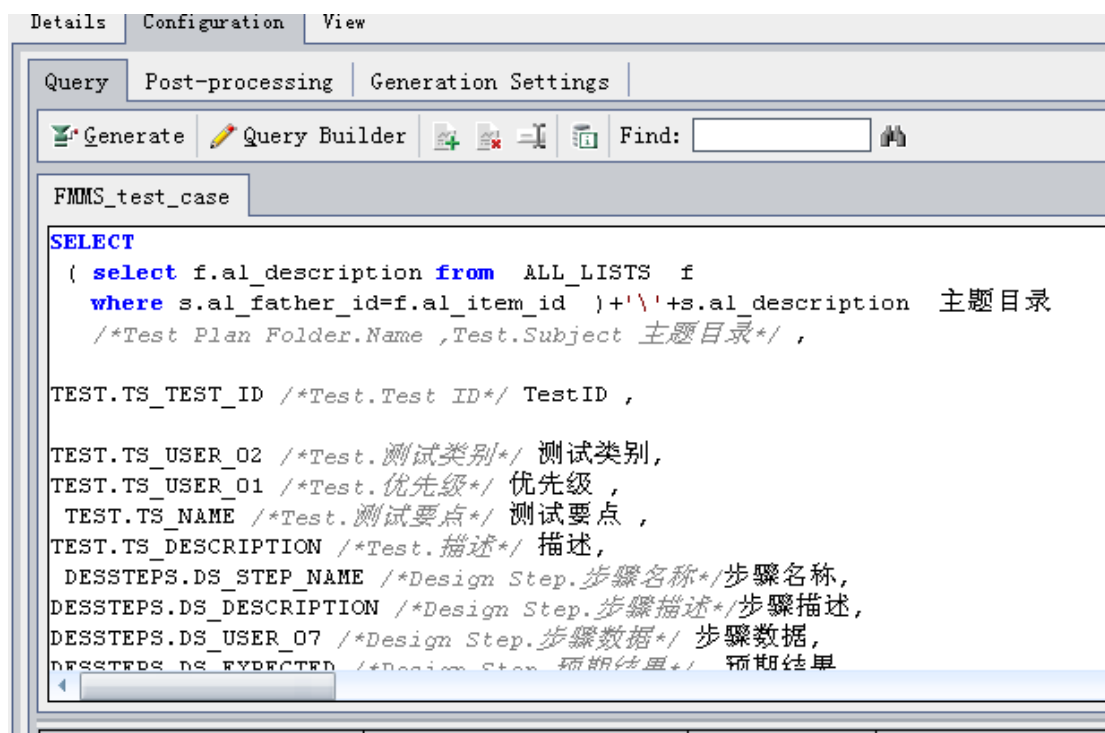


模块下，如下图操作进行创建 excel 报告：



3. 编写 sql 查询语句

对创建的报告，进行 sql 查询语句编写：



SQL 查询语句根据不同需要编写对应的语句，以下是根据几个情况分别给出对应的语句。其中 TEST.TS_USER_02 /*Test.测试类别*/ 测试类别, TEST.TS_USER_01 /*Test.优先级*/ 优先级, DESSTEPS.DS_USER_07 /*Design Step.步骤数据*/ 步骤数据, 是自定义的字段, 如果大家测试用例管理中没有对应字段信息, 可直接去掉。

a) 导出全部的 sql 语句

```

SELECT
( select f.al_description from ALL_LISTS f
  where s.al_father_id=f.al_item_id )+'\'+s.al_description 主题目录
/*Test Plan Folder.Name ,Test.Subject 主题目录*/ ,
TEST.TS_TEST_ID /*Test.Test ID*/ TestID ,
TEST.TS_USER_02 /*Test.测试类别*/ 测试类别,
TEST.TS_USER_01 /*Test.优先级*/ 优先级 ,
  TEST.TS_NAME /*Test.测试要点*/ 测试要点 ,
TEST.TS_DESCRIPTION /*Test.描述*/ 描述,
  DESSTEPS.DS_STEP_NAME /*Design Step.步骤名称*/步骤名称,
DESSTEPS.DS_DESCRIPTION /*Design Step.步骤描述*/步骤描述,
DESSTEPS.DS_USER_07 /*Design Step.步骤数据*/ 步骤数据,
DESSTEPS.DS_EXPECTED /*Design Step.预期结果*/ 预期结果,

TEST.TS_STATUS /*Test.状态*/ 状态,
TEST.TS_EXEC_STATUS /*Test.执行状态*/ 执行状态 ,
TEST.TS_RESPONSIBLE /*Test.设计者*/ 设计者
FROM
ALL_LISTS s /*Test Plan Folder*/ ,
TEST /*Test*/ ,
DESSTEPS /*Design Step*/

```

```
where
TEST.TS_SUBJECT=s.AL_ITEM_ID
and
TEST.TS_TEST_ID= DESSTEPS.DS_TEST_ID /*Design Step.Test ID*/
order by
主题目录 asc,s.al_description asc,TEST.ts_name asc, DESSTEPS.DS_STEP_ORDER asc
```

b) 按测试集导出 sql 语句

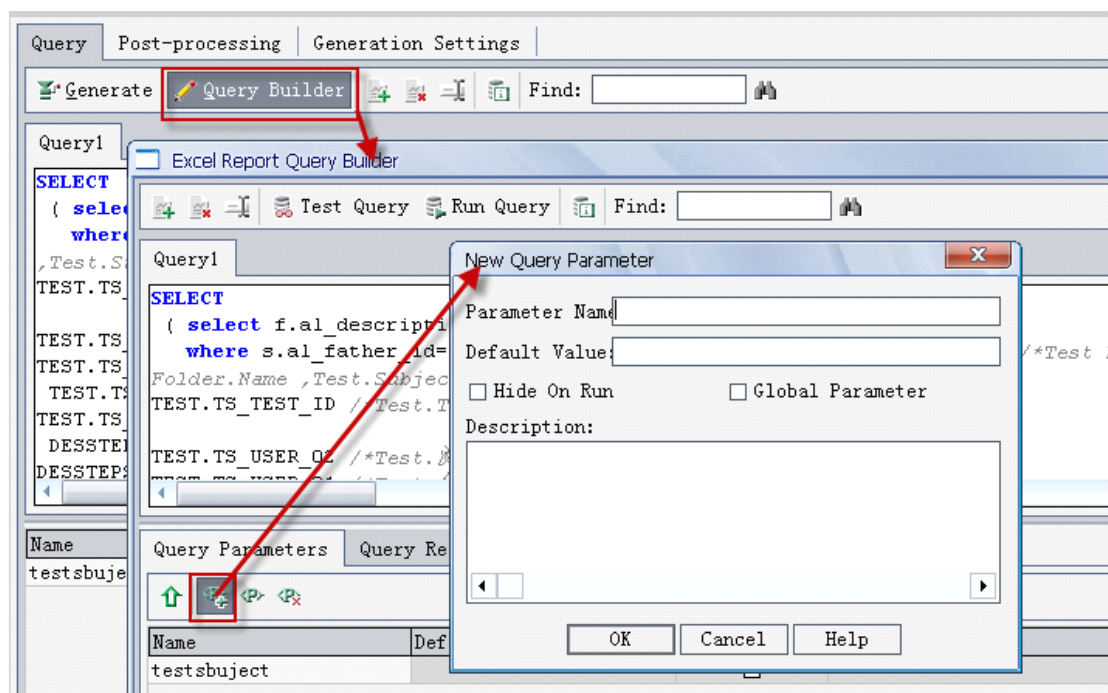
```
SELECT
( select f.al_description from ALL_LISTS f
where s.al_father_id=f.al_item_id )+'\'+s.al_description 主题目录 ,
TEST.TS_TEST_ID /*Test.Test ID*/ TestID ,

TEST.TS_USER_02 测试类别,
TEST.TS_USER_01 优先级 ,
TEST.TS_NAME 测试要点 ,
TEST.TS_DESCRIPTION 描述,
DESSTEPS.DS_STEP_NAME 步骤名称,
DESSTEPS.DS_DESCRIPTION 步骤描述,
DESSTEPS.DS_USER_07 步骤数据,
DESSTEPS.DS_EXPECTED 预期结果,
TEST.TS_RESPONSIBLE 设计者
FROM
ALL_LISTS s,
TEST ,
DESSTEPS
where
TEST.TS_SUBJECT=s.AL_ITEM_ID
and
TEST.TS_TEST_ID= DESSTEPS.DS_TEST_ID
and
TEST.TS_TEST_ID in (select TESTCYCL.TC_TEST_ID from TESTCYCL
where TESTCYCL.TC_CYCLE_ID
in (select CYCLE.CY_CYCLE_ID
from CYCLE
where CYCLE.CY_CYCLE like @testlab@
//'%自动生成%'
)
)

order by
s.al_father_id asc,s.al_description asc,TEST.ts_name asc , DESSTEPS.DS_STEP_ORDER
asc
```

注：@testlab@为参数，测试集名称

Query 的 querybuilder 页面进行 sql 语句的编辑与调试，还可添加设置参数。



c) 按测试树主题目录 sql 语句

SELECT

```
( select f.al_description from ALL_LISTS f
  where s.al_father_id=f.al_item_id )+'\'+s.al_description 主题目录 /*Test Plan
Folder.Name ,Test.Subject 主题目录*/ ,
TEST.TS_TEST_ID /*Test.Test ID*/ TestID ,
TEST.TS_USER_02 /*Test.测试类别*/ 测试类别,
TEST.TS_USER_01 /*Test.优先级*/ 优先级 ,
TEST.TS_NAME /*Test.测试要点*/ 测试要点 ,
TEST.TS_DESCRIPTION /*Test.描述*/ 描述,
DESSTEPS.DS_STEP_NAME /*Design Step.步骤名称*/步骤名称,
DESSTEPS.DS_DESCRIPTION /*Design Step.步骤描述*/步骤描述,
DESSTEPS.DS_USER_07 /*Design Step.步骤数据*/ 步骤数据,
DESSTEPS.DS_EXPECTED /*Design Step.预期结果*/ 预期结果,

//TEST.TS_STATUS /*Test.状态*/ 状态,
//TEST.TS_EXEC_STATUS /*Test.执行状态*/ 执行状态 ,
TEST.TS_RESPONSIBLE /*Test.设计者*/ 设计者
FROM
ALL_LISTS /*Test Plan Folder*/ s ,
TEST /*Test*/ ,
DESSTEPS /*Design Step*/
where
TEST.TS_SUBJECT=s.AL_ITEM_ID
and
```

```
TEST.TS_TEST_ID= DESSTEPS.DS_TEST_ID /*Design Step.Test ID*/
```

```
and s.AL_DESCRIPTION like @testsbuject@
```

```
// and ALL_LISTS.AL_DESCRIPTION in ( @testsbuject1@ , @testsbuject2@)
```

```
// and ALL_LISTS.AL_DESCRIPTION=@testsbuject@
```

```
order by 主题目录 asc,s.al_description asc,TEST.ts_name asc ,  
DESSTEPS.DS_STEP_ORDER asc
```

补充说明：

目前主要问题：

用例的主题目录可能无法完整显示。

目前排序的第一顺序按主题目录，但因主题目录不完整，所以会导致用例显示顺序不是很好。

考虑完善主题目录的命名规则:采用三级方式：

subject/主题为模块或菜单名/主题为菜单或功能名：测试要点

如果是子系统，可以考虑为四级

subject/**子系统/主题为模块菜单名/主题为菜单功能名：测试要点

即用例只有两个进行区分的主题目录节点。

4. Ecxcel 执行代码

（导出到 Excel 后运行的 Visual Basic 脚本 可控制 excel 数据显示）

```

Sub QC_PostProcessing()
Dim MainWorksheet As Worksheet
' Make sure your worksheet name matches!
Set MainWorksheet = ActiveWorkbook.Worksheets("Query1")
Columns(1).ColumnWidth = 16
Columns("e").ColumnWidth = 16
Columns(6).ColumnWidth = 20
Columns(8).ColumnWidth = 20
Columns(10).ColumnWidth = 30
' 设置列宽度
' Columns("b:d").AutoFit ' 列宽自动调整
Rows(1).Interior.ColorIndex = 34
' 标题行底纹颜色
Columns("e").WrapText = True
' 列内容自动换行
Columns("e:j").VerticalAlignment = xlTop
' 列内容居上显示

Dim DataRange As Range
Set DataRange = MainWorksheet.UsedRange

Dim Col As String: Col = "B"

```

```

Sub QC_PostProcessing()
Dim MainWorksheet As Worksheet
' Make sure your worksheet name matches!
Set MainWorksheet = ActiveWorkbook.Worksheets("FMMS_test_case")
Columns("e").ColumnWidth = 15
Columns(6).ColumnWidth = 20
Columns(8).ColumnWidth = 20
Columns(10).ColumnWidth = 30
'设置第 6,8,10 列的宽度
'Columns("b:d").AutoFit 'b 至 d 列宽自动调整
Rows(1).Interior.ColorIndex = 34 '标题行底纹颜色
Columns("e").WrapText = True 'e 列内容自动换行
Columns("e:j").VerticalAlignment = xlTop 'e 至 j 列内容居上显示

Dim DataRange As Range
Set DataRange = MainWorksheet.UsedRange

Dim Col As String: Col = "B" 'testid 列
Dim StartRow As Integer: StartRow = 2
Dim EndRow As Integer: EndRow = MainWorksheet.Range(Col & "65536").End(xlUp).Row
'Dim EndRow As Integer: EndRow = MainWorksheet.Range(Col & "65536").End(xlUp).Row

```

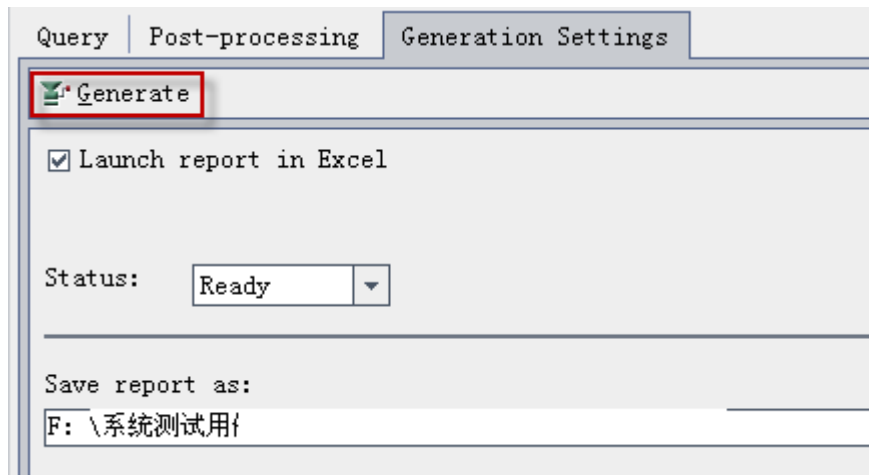
```
Dim i As Integer: i = StartRow
Dim j As Integer
With MainWorksheet
For j = 3 To EndRow
If Range(Col & i).Value = Range(Col & j).Value Then
.Range(.Cells(j, 1), .Cells(j, 6)).Value = ""
'相同测试下，重复内容为空显示
Else
i = j
End If
Next
j = j + 1
' Now that you have the data in DataRange you can process it.
End With
End Sub
```

5. Excel 设置

注意：运行后置处理脚本前，必须配置 Microsoft Excel 安全设置，以使 Excel 可以运行脚本。



6. Generate 生成文档



最后打开 excel 文件，如下效果图：

主题目录	TestID	测试类别	优先级	测试要点	描述	步骤名称	步骤描述	步骤数据	预期结果	设计
Subject\系统功能点测	16			系统时间的显示		Step 1	改变系统时间时间, 改变年份		系统时间是否正确显示。年份选项是否正确。是否含为历史年份, 当前年份, 明年年份	
						Step 2	页面时间的显示		默认显示当前系统时间日期。	
						Step 3	页面年份选项的显示		年份选项包含为历史年份, 当前年份, 明年年份 默认显示当前年份。(有特殊要求的, 根据需求显示,)	
Subject\系统功能点测	217			excel导出		Step 1	列表记录结果, 执行导出		对列表记录结果数据进行导出, 导出文件默认文件名合理, 可按修改的文件名保存(含空格等字符) 导出内容正确	
						Step 2	列表记录为空, 导出		空记录导出操作正常	
						Step 3	执行导出, 导出文件在路径中已存在该文件名		提示已存在, 是否覆盖: 是则覆盖, 否则不执行操作的 取消则可重新选择路径或输入文件名 已存在的文件打开时覆盖, 应能提示文件被使用, 无法覆盖, 请关闭文件在覆盖.	