

HP Defects and Requirements Exchange with HP Service Manager and HP Quality Center

Software Version: 1.00

Installation and Administration Guide

Document Release Date: June 2008

Software Release Date: June 2008

Last Updated: Jun 26, 2008



Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

© Copyright 2008 Hewlett-Packard Development Company, L.P.

Trademark Notices

Documentation Updates

This guide's title page contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates, or to verify that you are using the most recent edition of a document, go to:

http://ovweb.external.hp.com/lpe/doc_serv/

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

You can visit the HP software support web site at:

<http://support.openview.hp.com/support.jsp>

HP Software online support provides an efficient way to access interactive technical support tools. As a valued support customer, you can benefit by using the support site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract.

To find more information about access levels and HP Passport, go to:

http://support.openview.hp.com/new_access_levels.jsp

Contents

- 1 Introduction 9
 - Document Purpose 9
 - Synchronization Concepts 9
 - SM Change -> QC Defect 10
 - SM Change -> QC Requirement. 10
 - QC Defect -> SM Problem 10
 - Document Chapters 10
 - References. 11
- 2 Deployment Planning 13
 - Supported products 13
 - Deployment scenarios 14
 - Data types. 14
 - Deployment tasks. 14
 - Release package 15
- 3 SM Customization 17
 - Create Operators 17
 - Contact 17
 - Operator 17
 - Change Management 19
 - Create Profile Records 19
 - Add Fields 19
 - Specify Field External Access 20
 - Create Subform 21
 - Add Subform to Form 22
 - Add Format Control Calculations/Validations 22
 - Problem Management 24
 - Create Profile Records 24
 - Add Fields 24
 - Specify Field External Access 24
 - Create Subform 26
 - Add Subform to Form 26
- 4 QC Customization 29
 - Create an Integration Account 29
 - Create User 29
 - Create Group. 29
 - Assign Permission. 30
 - Defects 31

Add Fields	31
Add Tabs	31
Add Fields to Tabs	32
Create a View	33
Verify	34
Requirements	35
Add Fields	35
Add Tabs	36
Add Fields to Tabs	37
Create Folder "SM Incoming Changes"	37
5 QC Synchronizer Installation/Configuration	39
Install QCS	39
Extract Files	41
Deploy Adapters	41
Generate/Deploy Stub	42
Copy SM Adapter Configuration Files	42
Edit the Configuration Files	42
SM Change Management	43
SM Problem Management	44
6 Link Configuration	47
Link Configuration Basics	48
Create a Link	48
QC Field <-> SM Field mappings	51
Create Mapping	51
General Mapping Requirements	52
Matching Types	52
List Value Mappings	54
Constant -> SM field mappings	54
SM Change -> QC Defect	55
Specify Endpoints / Type of Link	55
Field Mappings	56
QC Field <-> SM Field	56
QC List Values <-> SM List Values (XML)	57
Events	59
Test	60
SM Change -> QC Requirement	61
Specify Endpoints / Type of Link	61
Field mappings	62
QC Field <-> SM Field	62
Events	63
Test	64
SM Problem to QC Defect	65
Specify Endpoints / Type of Link	65
Filters	65
Field Mappings	66

QC Field <-> SM Field.....	66
QC List Values <-> SM List Values (XML).....	67
Cosntants -> SM List Value Fields.....	69
Events	71
Test	72
A Known Issues	73
Timestamp	73
SM/QC Locked Record	73
Moving a Synchronized Requirement Stop Synchronization	74
QC 4-byte Integer Limitation	74
QC Date Field.....	74
QC Special Characters.....	74
Non-English Characters in a Synchronizer Report	74
Service Manager Fields with Non-English Names Not Supported.....	75
SM Integration User Account and Password Not Validated	75
B Error Messages	77
Installation	77
Configuration	78
Runtime	79
XML Validation	81

1 Introduction

This introduction describes

- *Document Purpose*
- *Synchronization Concepts*
- *Document Chapters*
- *References*

Document Purpose

This document describes how to configure and deploy the integration components

- Service Manager (SM)
- Quality Center (QC)
- Quality Center Synchronizer (QCS)

This document then describes how to configure and test synchronization links between QC and SM.



This document contains a lot of examples that use the SM and QC default installation configuration and databases. Your particular configuration may differ significantly. The example synchronization configuration may also differ significantly from your requirements.

The target readers include HP Consultants and/or Application Administrator who must set up and maintain QCS, ensuring that QCS meets all user organization procedural requirements. This document assumes that the reader is an experienced user of either (but not both) SM or QC, and therefore describes the basics of both SM and QC.

Synchronization Concepts

This section provides a detailed introduction to basic synchronization concepts.

- *SM Change -> QC Defect*
- *SM Change -> QC Requirement*
- *QC Defect -> SM Problem*

SM Change -> QC Defect

When a business owner enters a change request in SM and marks the change “Forward as defect”, a defect is created in QC. This informs the QA personnel that they should begin the QA process.

During the QA process, key information is synchronized from QC to SM. The integration administrator has the responsibility of determining the key information and specifying the information in the field mapping (using the provided integration tool). This allows the business owner to view updated (scheduled) information in the SM. The information includes the status of all changes in the testing cycle.

SM Change -> QC Requirement

The requirement synchronization features of SM and QC integration allow defects found during the change management process to be systematically tracked by SM and QC.

When a business owner enters a change request in SM and marks it as “Forward as requirement”, a requirement is created in QC. This informs the QA personnel that they should begin the QA process.

During the QA process, key information is synchronized from QC to SM. The integration administrator has the responsibility of determining the key information and specifying the information in the field mapping (using the provided integration tool). This allows the business owner to view updated (scheduled) information in SM.

QC Defect -> SM Problem

The business process for defect management in QC supports creation of known errors in SM based on information in QC. However, in the current solution, the integration can only create a problem in SM from a defect in QC. A user must create the known error in SM manually from the problem in SM. Known errors are a source of information for informal knowledge articles in the Knowledge Base.

Document Chapters

This document contains the following chapters:

- *Chapter 1, Introduction* (this chapter)
- *Chapter 2, Deployment Planning* describes the
 - Support products
 - Deployment scenarios. In general a single SM server can have links through a single QCS to multiple QC servers.
 - Deployment tasks for
 - QC (customizations)
 - SM (customizations)
 - QCS (installation / configuration)
 - Release package (directories and files)

- *Chapter 3, SM Customization* describes how to customize SM:
 - Create operators
 - Create fields
 - Allow external access to fields
 - Create subforms
 - Modify format control (change management)
- *Chapter 4, QC Customization* describes how to customize QC:
 - Create integration account
 - Create fields
 - Create tabs
 - Create defect view filter
 - Create requirements folder
- *Chapter 5, QC Synchronizer Installation / Configuration* describes how to
 - Install Synchronizer Server / Client
 - Generate/deploy stub/adapters
 - Create SM adapter configuration file
- *Chapter 6, Link Configuration* describes link configuration/testing for
 - SM Change -> QC Defect
 - SM Change -> QC Requirement
 - QC Defect -> SM Problem
- *Appendix A, Known Issues*
- *Appendix B, Error Messages*

References

- 1 HP QC Synchronizer User's Guide
- 2 HP Quality Center Administrator's Guide
- 3 HP Service Manager Installation Guide
- 4 HP Service Manager Online Help
- 5 Best Practices for Publishing and Consuming Web Services with ServiceCenter

2 Deployment Planning

This chapter describes deployment planning

- *Supported products*
- *Deployment scenarios*
- *Data types*
- *Deployment tasks*
- *Release package*

Supported products

Supported products are shown in the following table.

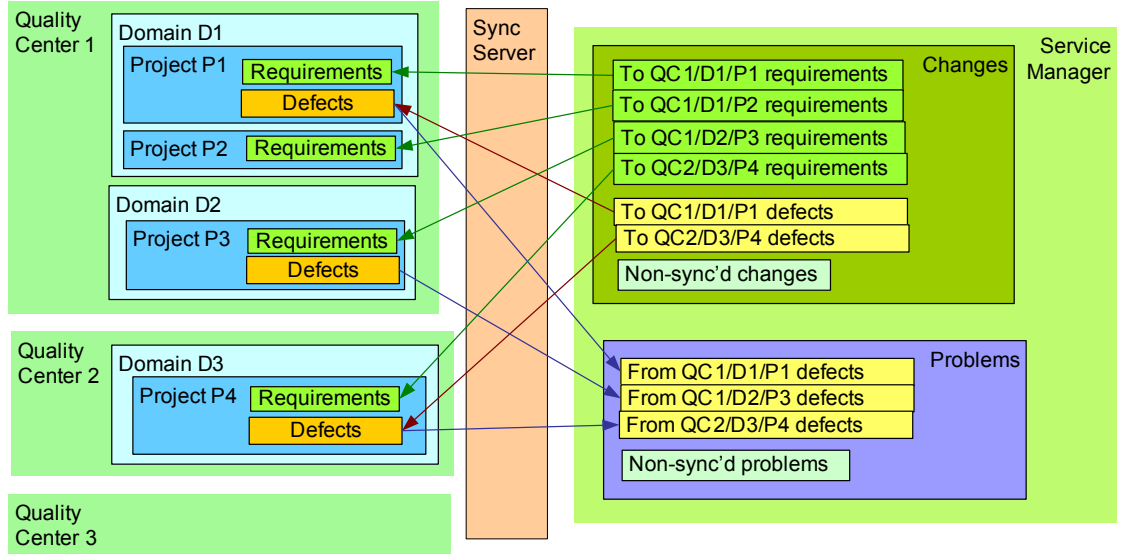
Platform	Version
QC Synchronizer	1.2
QC	9.2 patch 4
SM	7.01

Deployment scenarios

The following are the deployment scenario limitations:

- A single SM has a dedicated synchronizer.
- A single SM can connect to multiple QC's.

This is shown in the following diagram.



Data types

Matching Types on page 52 describes the data type requirements between QC, QCS, and SM.

Deployment tasks

Deployment tasks include:

- SM customization as described in *Chapter 3, SM Customization*.
- QC customization as described in *Chapter 4, QC Customization*.
- QC Synchronizer Server installation/configurations as described in *Chapter 5, QC Synchronizer Installation/Configuration*.
- Link configuration as described in *Chapter 6, Link Configuration*

Release package

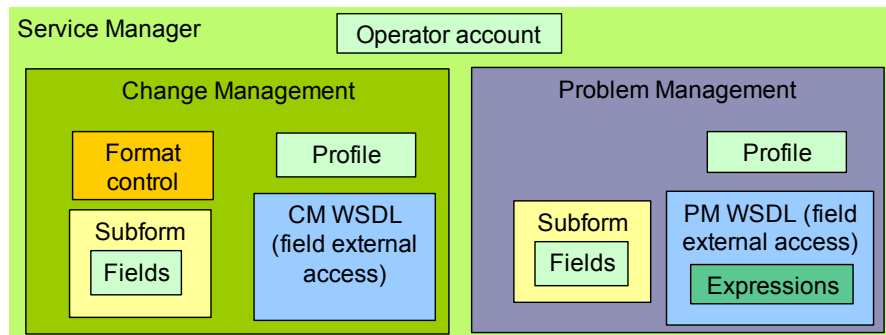
The release package is delivered as an executable self-extracting installer. Run the installer by double clicking `SMQC_Integration_xxx.exe` (where `xxx` is the version). The unzipped contents are shown in the following table.

Directory	Contents
sm-adapter\adapter	Adapter and dependencies (except stub)
sm-adapter\ant	Build lib
sm-adapter\bin	Script to generate the stub
sm-adapter\doc	Release documents (including this document)
sm-adapter\jdk5	Sun JDK 1.5
sm-adapter\lib	Binary libraries required to generate the stub
sm-adapter\sample	Examples of WSDL and adapter configuration
sm-adapter\tmp	Files for stub generation

3 SM Customization

To customize SM, follow these steps

- *Create Operators* on page 17
- *Change Management* on page 19
- *Problem Management* on page 24



Create Operators

Contact

Create a contact for the integration administrator (**Menu Navigation -> Basic System configuration -> Contacts**).

Page	Field	Value
Contact Information	Contact Name	<Administrator's name>
Contact Information	Full Name	<Administrator's full name>

Operator

The operator record identifies the logon name, password, and other settings for each SM operator. Create the required operator records (**Menu Navigation -> System Administration -> Ongoing Maintenance -> Operators**) with the parameters shown in the following table.

Table 1 Operator Record Parameters

No	Page	Field	Value	Comment
1	General	Logon Name	SMQCIntUser	
2	General	Full Name	QC Integration Default Account	
3	General	Contact ID	<Integration administrator's account in SM>	
4	General/ Application Profiles	Problem Profile	PMPProfile_QCInt	The problem management profile created in the following section.
5	General/ Application Profiles	Change Profiles	CMPProfile_QCInt	The change management profile created in the following section.
6	Security	Password	<Your password>	
7	Startup	Execute Capabilities	SOAP API	
8	Startup	Execute Capabilities	ChMAdmin	
9	Startup	Execute Capabilities	ProbAdmin	
10	Login Profile	Time Zone	Greenwich/ Universal (or create a timezone with no time difference or DST switch in Database Manager)	
11	Login Profile	Date Format	yy/mm/dd	The date format can not be changed (changing will cause loss of all data during synchronization).

Change Management

To customize SM for change management, follow these steps

- *Create Profile Records*
- *Add Fields*
- *Specify Field External Access*
- *Create Subform*
- *Add Subform to Form*
- *Add Format Control Calculations/Validations*

Create Profile Records

Profile records grant specific rights and privileges within a specific application (such as Service Desk or Incident Management) to SM operators.

- 1 Create a change management profile record (**Menu Navigation -> System Administration -> Ongoing Maintenance -> Profiles**) with the parameters shown in the following table.

No	Tab Page	Field	Value	Comment
1		Profile Name	CMProfile_QCInt	
2		Profile Area	Changes	
3	Security/Rights	Update	Always	
4	Security/Rights	View	Y	Check Box
5	Security/Rights	Reopen	Y	Check Box
6	Query	Query Options/All	Y	Check Box

Add Fields

- 2 Add the following fields to table cm3r (**System Definition -> Tables -> cm3r**). The values shown are required (do not change).

Field	Type
qcintegration.type	Character
qcintegration.id	Number
qcintegration.project	Character



The data type requirements for SM fields are described in *Matching Types* on page 52.

Specify Field External Access

- 3 Create a customized External Access Definition QCIntChangeService (**Menu Navigation > Tailoring > WSDL configuration**) with
 - Service Name: QCIntChangeService
 - Name: cm3r
 - Object Name: QCIntChange
 - Allowed Actions: Update

➤ The above values are required (do not change). The data type requirements for external access of SM fields is described in *Matching Types* on page 52.

This is shown in the following figure.

Allowed Actions	Action Names	Action Type
save	Update	

- 4 Export the required fields in the web service.

Field	Caption	Type
header, number	ChangeNumber	StringType
qcintegration.id	QCEntityID	IntType
sysmodtime	Modified	DateTimeType

➤ The caption value must be unique and alphanumeric (no spaces) with the first letter capitalized (“AValidCaption123”, “AnotherValidCaption”, and so on). The above values are required (do not change).

This is shown in the following diagram.

Field	Caption	Type
header, number	ChangeNumber	StringType
qcintegration.id	QCEntityID	IntType
header, priority. code	Priority	
severity	Urgency	StringType
sysmodtime	Modified	DateTimeType

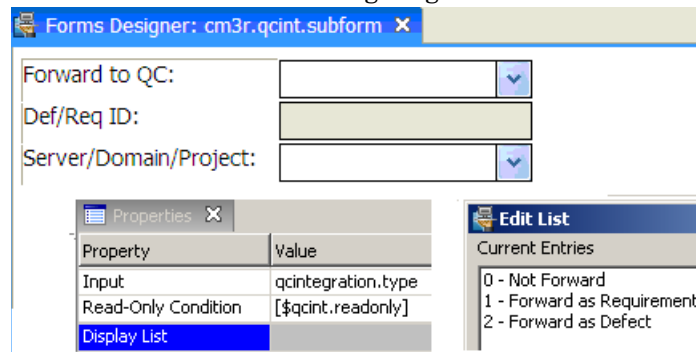
Create Subform

- 5 Create subform cm3r.qcint.subform without the Form Wizard (**Menu Navigation > Tailoring > Forms Designer**) with the following components on the canvas.

Table 2 Change Subform Parameters

Component	Properties
Label	Caption: "Forward to QC:"
Combo Box	<ul style="list-style-type: none"> • Input: "qcintegration.type" • Value List: "0;1;2" • Display List: "0 - Not Forward;1 - Forward as Requirement;2 - Forward as Defect" • Select Only: "Yes" • Read-Only Condition: "[\$qcint.type.readonly]"
Label	Caption: "Def/Req ID:"
Text	<ul style="list-style-type: none"> • Input: "qcintegration.id" • Read-Only: "Yes"
Label	Caption: "Server/Domain/Project:"
Combo Box	<ul style="list-style-type: none"> • Input: "qcintegration.project" • Value List: "server1/domain1/project1;server2/domain2/project2" • Read-Only Condition: "[\$qcint.project.readonly]" • Mandatory Condition: "[qcintegration.type]<>0"

This is shown in the following diagram.

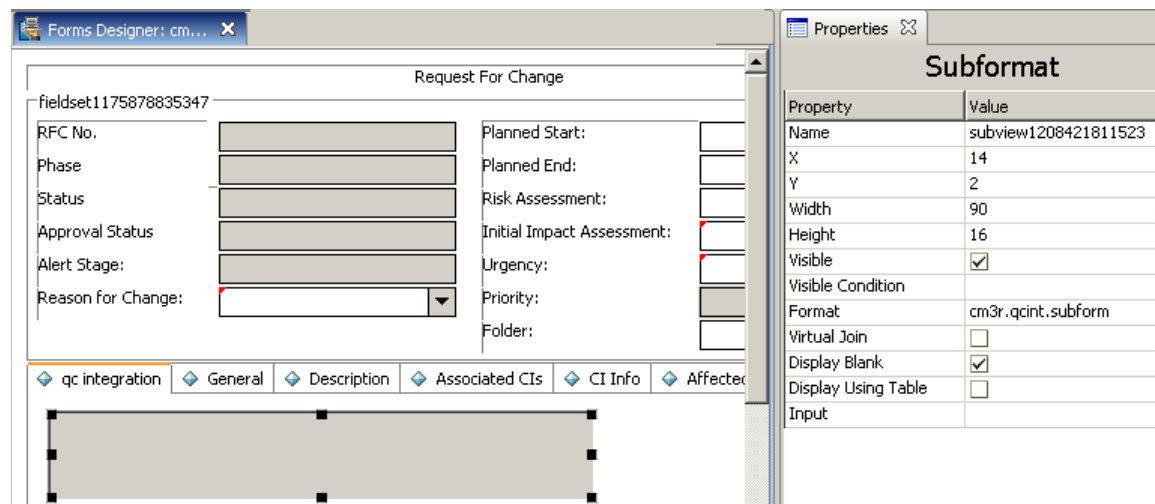


Add Subform to Form

To add a subform to a form, follow these steps.

- 6 Open the form of a phase of a category via the form designer (cm3r.rfc.build.g is used in this example).
- 7 Add a notebook tab with caption "QC Integration".
- 8 Add a subform to the new tab with format "cm3r.qcint.subform".

The following diagram shows the form.



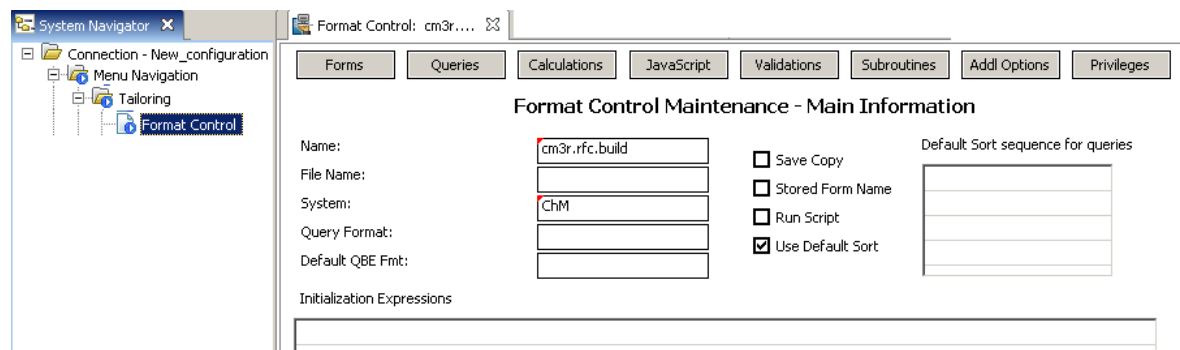
- 9 Save the changes.



If the error message "Format 'cm3r.qcint.subform' not found (display, show.rio)" appears, then restart the SM server to enable the subform.

Add Format Control Calculations/Validations

- 10 Open the form format control of the previous change form (in this example cm3r.rfc.build).



- 11 Click **Calculations**.

12 Add two records.

Parameter	Display	Initial	Calculation
1	true	true	<code>\$qcint.type.readonly=2;if (qcintegration.type in \$file~=0) then (\$qcint.type.readonly=1)</code>
2	true	true	<code>\$qcint.project.readonly=2;if (qcintegration.type in \$file~=0 and not null(qcintegration.project in \$file)) then (\$qcint.project.readonly=1)</code>

The change calculations are shown in the following diagram.

Forms Queries **Calculations** JavaScript Validations Subroutines Add Options Privileges

Format Control Maintenance - Calculations

Name: cm3r.rfc.build View: short

add	update	delete	disp...	initial	calculation
					risk.assessment in \$file=nullsub(risk.assessment in \$file, "1")
true	true				misc3 in \$file=nullsub(misc3 in \$file, "no")
		priorit...			if (misc3 in \$file="yes") then (\$phasepntr=3;current.phase in \$file="RFC Testing")
		true			billtype in \$file=nullsub(billtype in \$file, "dept")
			true	true	\$qcint.readonly=2;if (qcintegration.type in \$file="1" or qcintegration.type in \$file="2") then (\$qcint.readonly=1)

13 Click **Validations**.

14 Add one record with the following parameters.

Table 3 Change Validation Record

No	Parameter	Value
1	Validation	<code>not null(qcintegration.project in \$file)</code>
2	Message	The Server/Domain/Project is required.
3	Add	<code>qcintegration.type in \$file~=0</code>
4	Update	<code>qcintegration.type in \$file~=0</code>
5	Set Focus to	<code>qcintegration.project</code>

The change validation record is shown in the following diagram.

Calculations JavaScript **Validations** Subroutines

Format Control Maintenance - Validations

Name: cm3r.rfc.build View: long

Use Pop-up messages:

Validations

Validation	<code>not null(qcintegration.project in \$file)</code>	Delete	<input type="text"/>
Message	The Server/Domain/Project is required.	Display	<input type="text"/>
Comments		Initial	<input type="text"/>
Add	<code>qcintegration.type in \$file~=0</code>	Set Focus to	<code>qcintegration.project</code>
Update	<code>qcintegration.type in \$file~=0</code>	Message ID	<input type="text"/>

15 Save the changes.

Problem Management

To customize problem management, follow these steps.

- *Create Profile Records*
- *Add Fields*
- *Specify Field External Access*
- *Create Subform*
- *Add Subform to Form*

Create Profile Records

- 1 Create a problem management profile record (**Menu Navigation -> System Administration -> Ongoing Maintenance -> Profiles**) with the parameters shown in the following table.

No	Tab Page	Field	Value	Memo
1		Profile Name	PMProfile_QCInt	
2	Problems/Security/Rights	New	Y	Check Box
3	Problems/Security/Rights	Close	Y	Check Box
4	Problems/Security/Rights	Update	Always	
5	Problems/Security/Rights	Reopen	Y	Check Box

Add Fields

- 2 Add the following fields to the table "rootcause" (**Menu Navigation > Tailoring > Database Dictionary**). The values shown are required (do not change).

Field	Type
qcintegration.id	Number
qcintegration.project	Character



The data type requirements for SM fields are described in *Matching Types* on page 52.

Specify Field External Access

- 3 Create a customized External Access Definition QCIntProblemService (**Menu Navigation > Tailoring > WSDL configuration**).
 - Service Name: QCIntProblemService
 - Name: rootcause
 - Object Name: QCIntProblem

- Allowed Action: add / Create
- Allow Action: save / Update



The above values are required (do not change). The data type requirements for external access of SM fields is described in *Matching Types* on page 52.

This is shown in the following diagram.

External Access Definition

Service Name:

Name: Object Name:

Allowed Actions Expressions Fields

Allowed Actions	Action Names	Action Type
add	Create	
save	Update	

4 Export the required fields in the web service.

Field	Caption	Type
qcintegration.id	QCEntityID	IntType
id	ProblemID	StringType
sysmodtime	Modified	DateTimeType
qcintegration.project	QCProject	StringType



The caption value must be unique and alphanumeric (no spaces) with the first letter capitalized (“AValidCaption123”, “AnotherValidCaption”, an so on). The above values are required (do not change).

This is shown in the following diagram.

External Access Definition

Service Name:

Name:

Allowed Actions Expressions Fields

Field	Caption	Type
qcintegration.id	QCEntityID	IntType
id	ProblemID	StringType
sysmodtime	Modified	DateTimeType
qcintegration.project	QCProject	StringType
incident.category	Category	StringType

5 Create the following expressions to fulfill the business logic requirements of problem management.

No	Expression
1	cleanup(\$pm.activity);cleanup(\$rc.update);if same(update in \$L.file, update in \$L.file.save) then (\$L.need.to.update=true)
2	\$rc.update=update in \$L.file;if (denu(\$rc.update)={}) then (\$rc.update={"no update provided"})
3	if (\$L.need.to.update=true) then (\$rc.update={"no update provided"})

No	Expression
4	update in \$L.file=update in \$L.file.save
5	current.phase in \$L.file="XXX" (replace XXX with a phase name, such as 'Problem Identification and Classification').
6	category in \$L.file="ITIL" (only apply for the demo data of SM 7.0.1).

This is shown in the following diagram.

External Access Definition

Service Name:

Name: Object Name:

◆ Allowed Actions ◆ Expressions ◆ Fields

Expressions

```
cleanup($pm.activity);cleanup($rc.update);if same(update in $L.file, update in $L.file.save) then ($L.need.to.update=true)
$rc.update=update in $L.file;if (dnull($rc.update)=()) then ($rc.update=("no update provided"))
if ($L.need.to.update=true) then ($rc.update=("no update provided"))
update in $L.file=update in $L.file.save
current.phase in $L.file="Problem Investigation and Diagnosis"
category in $L.file="ITIL"
```

Create Subform

- 6 Create subform pm.qcint.subform without the Form Wizard (**Menu Navigation > Tailoring > Forms Designer**) with the following components.

Table 4 Problem Subform Parameters

Component	Properties
Label	Caption: "Defect ID:"
Text	Input: "qcintegration.id" Read-Only: "Yes"
Label	Caption: "Server/Domain/Project:"
Text	Input: "qcintegration.project" Read-Only: "Yes"

This is shown in the following diagram.

Forms Designer: pm.qcint.subform

Defect ID:

Server/Domain/Project:

Properties

Property	Value
Name	input1206947870453
Input	qcintegration.id

Add Subform to Form

- 7 Open the default form of one phase of Problem Management via Form Designer (in this example PM.pc.ident.and.class).
- 8 Add a Notebook Tab with caption "QC Integration".

- 9 Add a Subform to the new tab with format "pm.qcint.subform".

The screenshot shows the Forms Designer interface. The main window title is '*Forms Designer: PM.pc.ident.and.class'. The subform is titled 'Subformat' and has the following properties:

Property	Value
Format	pm.qcint.subform
Virtual Join	<input type="checkbox"/>
Display Blank	<input checked="" type="checkbox"/>
Display Using Table	<input type="checkbox"/>
Input	

- 10 Save the changes.

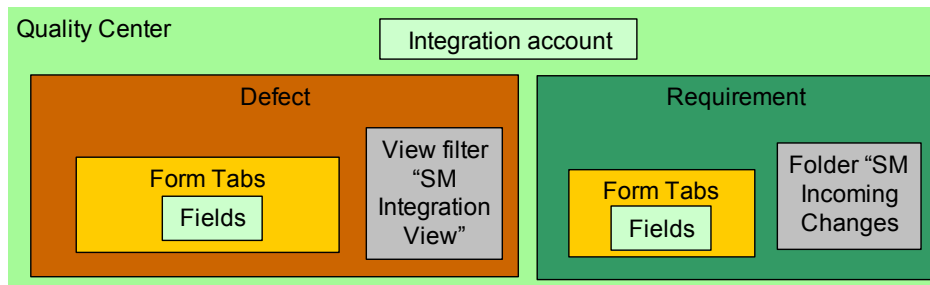


If the error message "'Format 'pm.qcint.subform' not found (display, show.rio)" appears, then restart the SM server to enable the subform.

4 QC Customization

To customize QC, follow these steps.

- *Create an Integration Account* on page 29
- *Defects* on page 31
- *Requirements* on page 35



Create an Integration Account

To create an integration account, follow these steps.

- *Create User*
- *Create Group*
- *Assign Permission*

Create User

To create a user, follow these steps.

- 1 Log in to the "Quality Center - Administrator" account.
- 2 In the **Site Users** tab, create and configure integration account **SMQCIntUser** (including the full name and password).
- 3 In the **Site Projects** tab, choose the project from the list.
- 4 Click the **Project Users** tab in the right panel and click **Add From The Users List**.
- 5 Add the configured user **SMQCIntUser** to the project.
- 6 Log off.

Create Group

- 7 Log on to the QC administrator account.

- 8 Click **TOOLS->Customize...**
- 9 Select **Groups**.
- 10 Click **New**.
- 11 Enter name **SMIntegration**.
- 12 For Create As: select **Viewer**.
- 13 Click **OK**.
- 14 Select **Yes** to create the user group.

Assign Permission

- 15 Click **Change** and assign permissions for the user group as shown in the following table.

Requirement	Add Requirement/ Modify Requirement	<input checked="" type="checkbox"/> Add Requirement <input checked="" type="checkbox"/> Modify Requirement <input type="checkbox"/> Delete Requirement <input type="checkbox"/> Add Tests To Coverage <input type="checkbox"/> Remove Tests From Coverage <input type="checkbox"/> Add Requirement Traceability <input type="checkbox"/> Modify Requirement Traceability <input type="checkbox"/> Remove Requirement Traceability <input type="checkbox"/> Risk-Based Quality Management
Defects	Add Defects/Modify Defects	<input checked="" type="checkbox"/> Add Defect <input checked="" type="checkbox"/> Modify Defect <input type="checkbox"/> Delete Defect <input type="checkbox"/> Add Defect Link <input type="checkbox"/> Modify Defect Link <input type="checkbox"/> Remove Defect Link

- 16 Add the integration user **SMQCIntUser** to group **SMIntegration**.
- 17 Save and close. The integration account is created.

Defects

This section describes the required Defects customizations:

- [Add Fields](#)
- [Add Tabs](#)
- [Add Fields to Tabs](#)
- [Create a View](#)

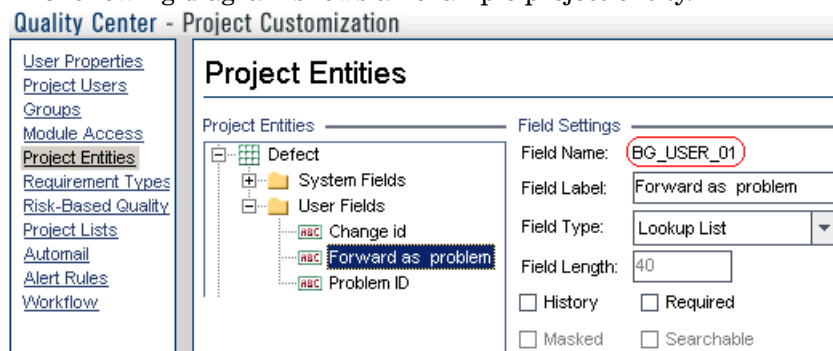
Add Fields

To add the required fields for defect customization, follow these steps.

- 1 Log on to QC as the project administrator.
- 2 Click **Tools / Customize**. Module "QC - Project Customization" appears.
- 3 Add the following fields for the defect entity in project entities (XX YY XZ are sequential numbers auto-generated by QC).

Field Name	Field Label	Field Type
BG_USER_XX	Forward as Problem	Lookup List/YesNo
BG_USER_XY	Change ID	String
BG_USER_XZ	Problem ID	String

The following diagram shows an example project entity.



The data type requirements for QC fields is described in [Matching Types](#) on page 52.

Add Tabs

To add tabs to the defect form and show fields on these tabs, follow these steps.

- 4 In "QC - Project Customization" click **Workflow**.
- 5 Click **Workflow -> Script Editor**.

6 Choose **Defects module script**.

Quality Center - Project Customization

7 Add the following code to the GetNewBugPageName event procedure (which is triggered before QC opens the Add Defect dialog box).

```
select case PageNum
  case "2"
    GetNewBugPageName = "SM Integration (New)"
  end select
```

► “2” specifies tab 2 (the second tab). For a new bug, the tab name is "SM Integration (New)".

8 Add the following code to the GetDetailsPagename event procedure (which is triggered before QC displays Defect Details dialog box).

```
select case PageNum
  case "2"
    GetDetailsPageName = "SM Integration (Details)"
  end select
```

► “2” specifies tab 2 (the second tab). For an existing defect, the tab name is "SM Integration (Details)".

Add Fields to Tabs

9 If WizardFieldCust_Details and WizardFieldCust_Add are not found in the list, then do the following to generate these two methods.

1. **Script Generator - Add Defect Field Customization**
2. **Script Generator - Defect Details Field Customization**

10 Add the following code to the WizardFieldCust_Details event procedure.

```
SetFieldApp "BG_USER_XX", True, False, 1, 0
```



```
SetFieldApp "BG_USER_XY", True, False, 1, 1
SetFieldApp "BG_USER_XZ", True, False, 1, 2
```

The parameters are

- Field name (BG_USER_XX, where XX = 2 digits)
- Visible (True)
- Required (False)
- Page number (start from 0)
- View order (start from 0)

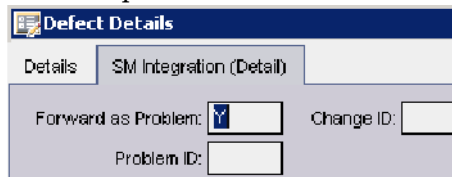
- 11 Add the following code to the WizardFieldCust_Add event procedure.

```
SetFieldApp "BG_USER_XX", True, False, 1, 0
SetFieldApp "BG_USER_XY", True, False, 1, 1
SetFieldApp "BG_USER_XZ", True, False, 1, 2
```

- 12 Set the Readonly fields by adding the following lines to Bug_New and Bug_Moveto subroutines:

```
if (Bug_Fields("BG_USER_XX").Value="Y") then
    Bug_Fields("BG_USER_XX").IsReadOnly=True
end if
Bug_Fields.Field("BG_USER_XY").IsReadOnly=True
Bug_Fields.Field("BG_USER_XZ").IsReadOnly=True
```

The if loop above marks the field "Forward as Problem" as read only after selected and saved.

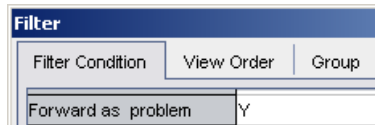


- 13 Exit customization (save changes).

Create a View

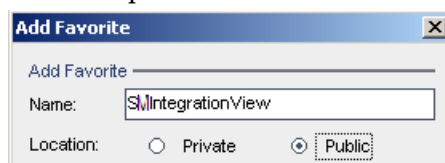
- 14 In the Defects module, click **View / Filter/Sort / Set Filters/Sort**. The purpose of this view is to let QC Synchronizer correctly filter those defects to be synchronized to SM as problems.

- 15 Set **Forward as Problem** to **Y**.



- 16 Add a view to Favorites:

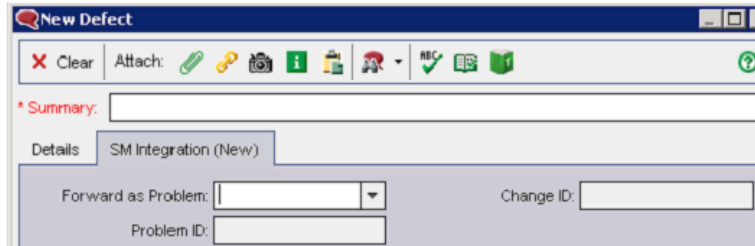
- Name: SMIntegrationView
- Location: public



In the QC Synchronizer this view will be selected as the QC data filter. QC defects can not be forwarded to SM without the filter.

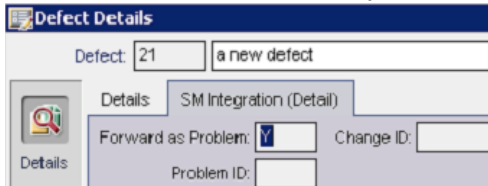
Verify

- 17 Create a new defect. The dialog box should have a new tab "SM Integration (New)" with 3 fields.



The screenshot shows a window titled "New Defect". At the top, there is a toolbar with icons for "Clear", "Attach", and various file operations. Below the toolbar is a "Summary" field. The main area has two tabs: "Details" and "SM Integration (New)". The "SM Integration (New)" tab is active and contains three fields: "Forward as Problem:" with a dropdown menu, "Change ID:" with a text input field, and "Problem ID:" with a text input field.

- 18 When opening an existing defect, if "Forward as Problem" have been set previously to "Y", then this field will be readonly.



The screenshot shows a window titled "Defect Details". At the top, there is a "Defect:" field with the value "21" and a text input field containing "a new defect". Below this is a "Details" section with two tabs: "Details" and "SM Integration (Detail)". The "SM Integration (Detail)" tab is active and contains three fields: "Forward as Problem:" with a dropdown menu showing "Y", "Change ID:" with a text input field, and "Problem ID:" with a text input field.

Requirements

To customize QC for requirements management, follow these steps

- *Add Fields*
- *Add Tabs*
- *Add Fields to Tabs*
- *Create Folder "SM Incoming Changes"*

Add Fields

19 Add a field for the requirement entity in Project Entities.

Field Name	Field Label	Field Type
RQ_USER_ZZ	Change ID in SM	String



The data type requirements for QC fields is described in *Matching Types* on page 52. ZZ is a sequential number auto-generated by the QC system.

This is shown in the following diagram.

20 In Requirement Types add the field "Change ID in SM" to the Business type requirement. "Business" type is the default requirement type for incoming requirements (other types can be used).

Add Tabs

To add tabs to the requirement form and display the fields on these tabs, follow these steps.

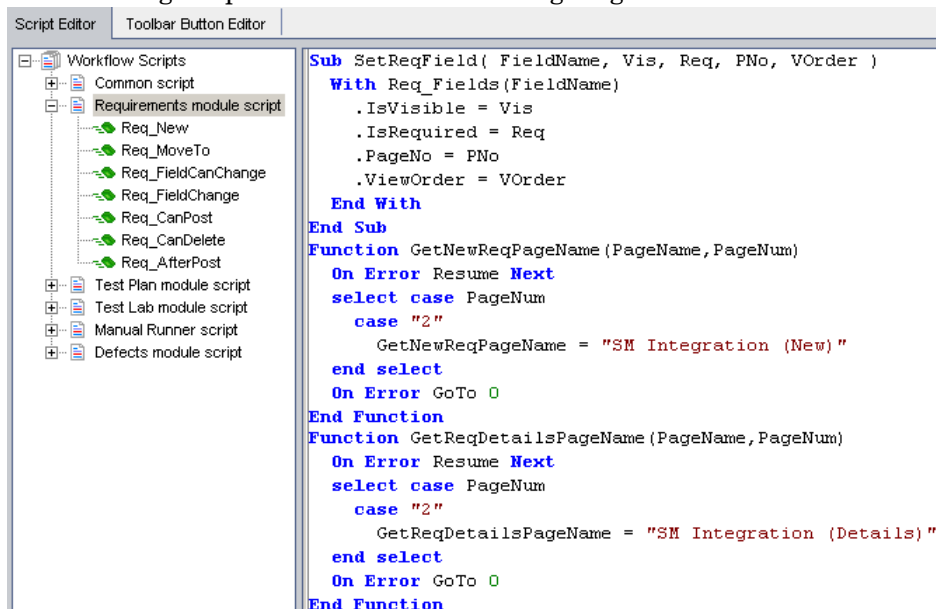
21 Click **Workflow-> Script Editor**. Add the following code to the requirement module.



For a new requirement, the tab name is "SM Integration (New)". For an existing requirement, the tab name is "SM Integration (Details)". "2" specifies tab 2 (the second tab). If N tabs exist, then the number of a new tab should be N+1. This function is called when an existing requirement is shown in the dialog box.

```
Sub SetReqField( FieldName, Vis, Req, PNo, VOrder )
    With Req_Fields(FieldName)
        .IsVisible = Vis
        .IsRequired = Req
        .PageNo = PNo
        .ViewOrder = VOrder
    End With
End Sub
Function GetNewReqPageName (PageName, PageNum)
    On Error Resume Next
    select case PageNum
        case "2"
            GetNewReqPageName = "SM Integration (New)"
    end select
    On Error GoTo 0
End Function
Function GetReqDetailsPageName (PageName, PageNum)
    On Error Resume Next
    select case PageNum
        case "2"
            GetReqDetailsPageName = "SM Integration (Details)"
    end select
    On Error GoTo 0
End Function
```

The resulting script is shown in the following diagram.



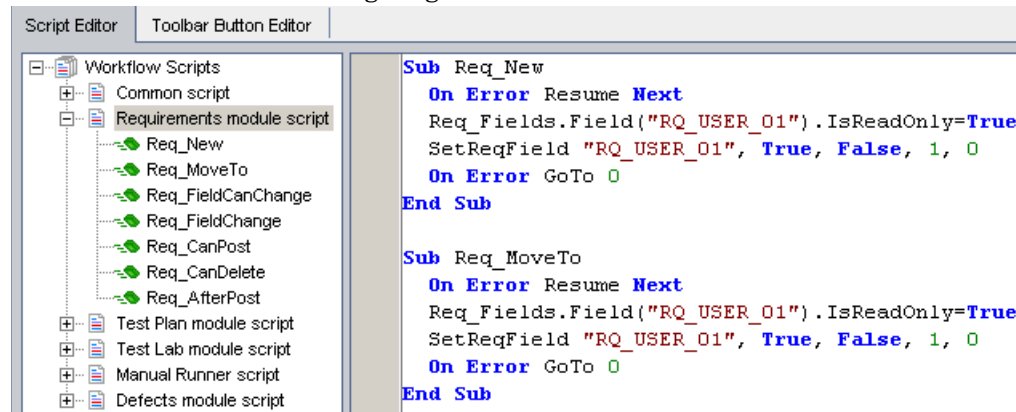
Add Fields to Tabs

To set the fields as read only and place the fields on the tabs, follow these steps.

- 22 In the Script Editor for the requirements module script, add the following code to Req_New and Req_MoveTo (Req_New is called when a new requirement is created; Req_MoveTo is called when an existing requirement is opened).

```
Req_Fields.Field("RQ_USER_ZZ").IsReadOnly=True  
SetReqField "RQ_USER_ZZ", True, False, 1, 0
```

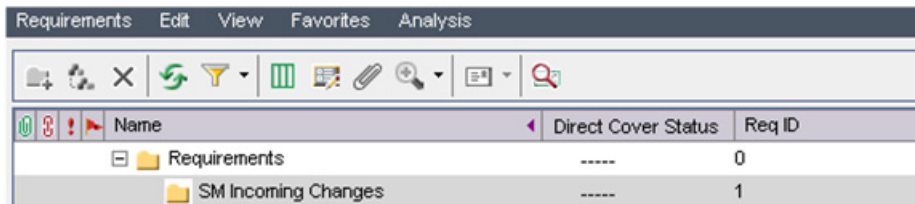
This is shown in the following diagram.



Create Folder "SM Incoming Changes"

To create the folder for the requirements created from SM changes, follow these steps.

- 23 From the menu, select **Requirements / New Folder**.
- 24 Set the folder name to **SM Incoming Changes**.



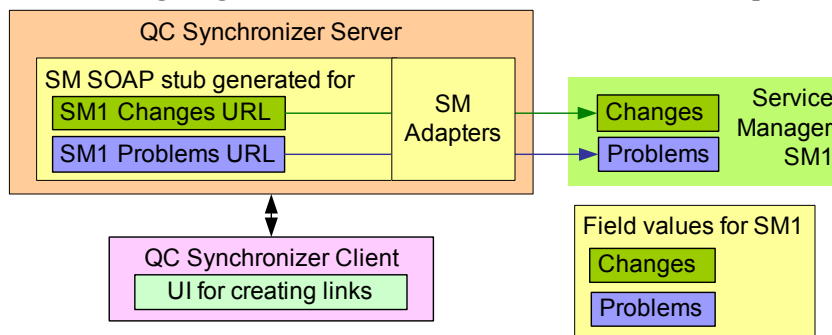
5 QC Synchronizer Installation/Configuration

The QC Synchronizer (QCS) allows centralized management of a set of tightly coupled one-to-one data synchronization links and provides an open and extensible platform for the development of new data synchronization adapters to entity repositories.

To install and configure the synchronizer, follow these steps.

- *Install QCS*
- *Extract Files*
- *Deploy Adapters*
- *Generate / Deploy Stub*
- *Copy SM Adapter Configuration Files*
- *Edit the Configuration Files*

The following diagram shows an overview of the installation process.



Install QCS

QCS is available from

<http://updates.merc-int.com/qualitycenter/qc90/sync/qcsynchronizer/index.html>

- If there is need to remove previously installed QCS, make sure the postgres database installation folder and the OS user "postgres" deleted.
- After installing QCS server, QC client should also be installed in the same machine to let QC synchronizer properly connect to QC.

Refer to the QC Synchronizer user guide for installation instructions (QCSyncUG.pdf). You can get the user guide from the installation package or from

http://ovweb.external.hp.com/lpe/doc_serv/



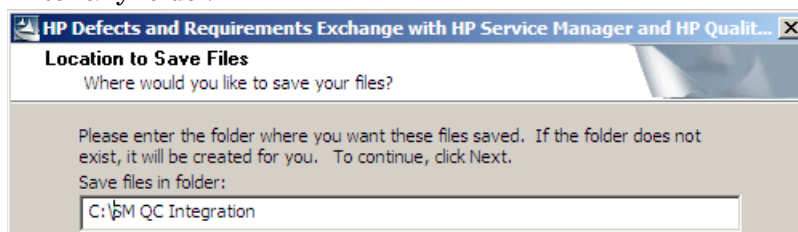
After the installation check if QCS local time is the same with SM. For example, if the UTC time on SM is 2008-12-12 13:00:00, then the UTC time on QCS should be the same (or at least within 5 minutes). If the UTC time on QCS is 2008-12-12 13:06:00 then the synchronization may lose data from SM.

Extract Files

- 1 Double-click on **Smqc_integration_01.00.000.exe**. The Welcome dialog appears.



- 2 Click **Next**. The “Location to Save Files” dialog appears.
- 3 Enter any folder.



- 4 Click **Next**. The files are extracted. A web page opens in a browser with links to the user guide and release notes.



- 5 Click **Explore Package**. The contents of the release package are shown.

Deploy Adapters

- 6 Copy all files under [release-package]\adapter to c:\Program Files\HP\Quality Center Synchronizer\adapters\lib. Adapters include
 - sm-adapter-01.00.000.jar
 - sm-adapter-axis-1.4.jar
 - sm-adapter-commons-discovery-0.2.jar
 - sm-adapter-commons-lang-2.3.jar
 - sm-adapter-jaxrpc-1.1.jar
 - sm-adapter-jdom-1.1.jar
 - sm-adapter-saaj-1.2.jar
 - sm-adapter-wsdl4j-1.5.1.jar

Generate/Deploy Stub

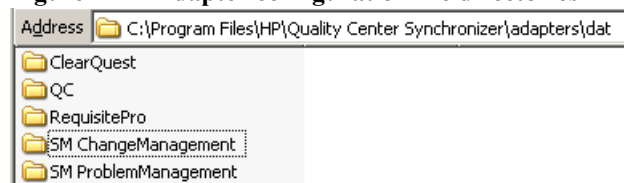
- 7 Start the SM service (stub generation requires access to SM).
- 8 Edit following lines in [release-package]\bin\build.properties as required for access to SM:

```
#Set up WSDL URL, please change the URL to your actual SM server, eg,  
http://<your-server>:<port>/.../<service-name>.wsdl  
sm.change.wsdl=http://localhost:13080/sc62server/PWS/  
QCIntChangeService.wsdl  
sm.problem.wsdl=http://localhost:13080/sc62server/PWS/  
QCIntProblemService.wsdl
```
- 9 Run the script build.bat from command line (check the console output for errors). The stub [release-package]\build\sm-adapter-ws-client.jar is generated.
- 10 Copy the stub to C:\Program Files\HP\Quality Center Synchronizer\adapters.

Copy SM Adapter Configuration Files

- 11 Start/restart the QCS service. The directories “C:\Program Files\HP\Quality Center Synchronizer\adapters\dat\SM ChangeManagement” and “SM ProblemManagement” appear after the synchronizer service starts (this can take up to a minute).

Figure 1 Adapter configuration file directories



- 12 Copy the file [release-package]\sample\configuration_file_default.xml to the following folders
 - “SM ChangeManagement”
 - “SM ProblemManagement”

Edit the Configuration Files

- 13 Edit the files as described below. The files will later be specified when creating links.

The configuration file is an XML file that provides Change/Problem field values to the SM adapter. These values include

- Field name (the caption of a field in SM WSDL configuration form, such as "Status", "Priority")
- List type (array (multi-value list) or single-value list)

- For a value list, the mapping of the value in the database and the exposed caption (for a type other than a value list type, the adapter automatically determines the desired data type).

Below are two sample configuration files. The supported data types are

- String
- Number
- Date
- Single_Value_List
- Multi_Value_List

The default field configuration is readable and writable with unlimited length. But it requires explicit specification for a Single_Value_List or a Multi-Value_List. It also needs explicit specification on read-only type of field.

Note:

- You must specify the type and read/write explicitly only for a Single_Value_List/Multi_Value_List.
- If a field is not configured, then the field is read/write with unlimited length.
- A list or multi-list field must contain item elements. For each item specify the value and display text in the form `<itg:item value="$value">$display text</itg:item>`.
- If the field in WSDL is an Array then it must be mapped to Multi_Value_List.
- If the QC field is User_List, then you can only specify String or Single_Value_List for the corresponding SM field.
- If the field is read only you must set the attribute **readonly** as **true**.
- If the field attribute **required** is **mandatory**, then the field is mandatory for creation of a new entity.
- If the field has a length limitation (attribute **length**). Values from other endpoints will be truncated to match this limitation.

SM Change Management

The following is typical for SM change management.

```
<?xml version="1.0" encoding="UTF-8"?>
<itg:mapping xmlns:itg="http://www.hp.com/smci/SMQCIntegration/config">
<itg:module name="change">
  <itg:field name="Urgency" tye="Single_Value_List" readonly="false"
required="mandatory" length="50">
    <itg:items>
      <itg:item value="1">1 - Critical</itg:item>
      <itg:item value="2">2 - High</itg:item>
      <itg:item value="3">3 - Average</itg:item>
    </itg:items>
  </itg:field>
</itg:module>
```

SM Problem Management

The following is the included configuration_file_default.xml for SM problem management.

```
<?xml version="1.0" encoding="UTF-8"?>
<itg:mapping xmlns:itg="http://www.hp.com/smci/SMQCIntegration/config">

<itg:module name="problem">
  <itg:field name="Status" type="Single_Value_List"
required="mandatory">
    <itg:items>
      <itg:item value="Open">Open</itg:item>
      <itg:item value="Accepted">Accepted</itg:item>
      <itg:item value="Work In Progress">Work In Progress</itg:item>
      <itg:item value="Pending Vendor">Pending Vendor</itg:item>
      <itg:item value="Pending User">Pending User</itg:item>
      <itg:item value="Rejected">Rejected</itg:item>
      <itg:item value="Deferred">Deferred</itg:item>
    </itg:items>
  </itg:field>
<itg:field name="AssignmentGroup" type="Single_Value_List"
required="mandatory">
    <itg:items>
      <itg:item value="Application">Application</itg:item>
      <itg:item value="Network">Network</itg:item>
    </itg:items>
  </itg:field>
<itg:field name="Service" type="Single_Value_List" required="mandatory">
    <itg:items>
      <itg:item value="Applications">Applications</itg:item>
      <itg:item value="Service Management">Service Management</itg:item>
    </itg:items>
  </itg:field>
<itg:field name="Title" type="String" required="mandatory" length="50"/>
  <itg:field name="Description" type="String" required="mandatory"/>
<itg:field name="Area" type="Single_Value_List" required="mandatory">
    <itg:items>
      <itg:item value="data">data</itg:item>
    </itg:items>
  </itg:field>
<itg:field name="Subarea" type="Single_Value_List" required="mandatory">
    <itg:items>
      <itg:item value="data or file corrupted">data or file corrupted</
itg:item>
    </itg:items>
  </itg:field>
  <itg:field name="Impact" type="Single_Value_List" readonly="false"
required="mandatory">
    <itg:items>
      <itg:item value="1">1 - Enterprise</itg:item>
      <itg:item value="2">2 - Site/Dept</itg:item>
      <itg:item value="3">3 - Multiple Users</itg:item>
      <itg:item value="4">4 - User</itg:item>
    </itg:items>
  </itg:field>
</itg:module>
</itg:mapping>
```

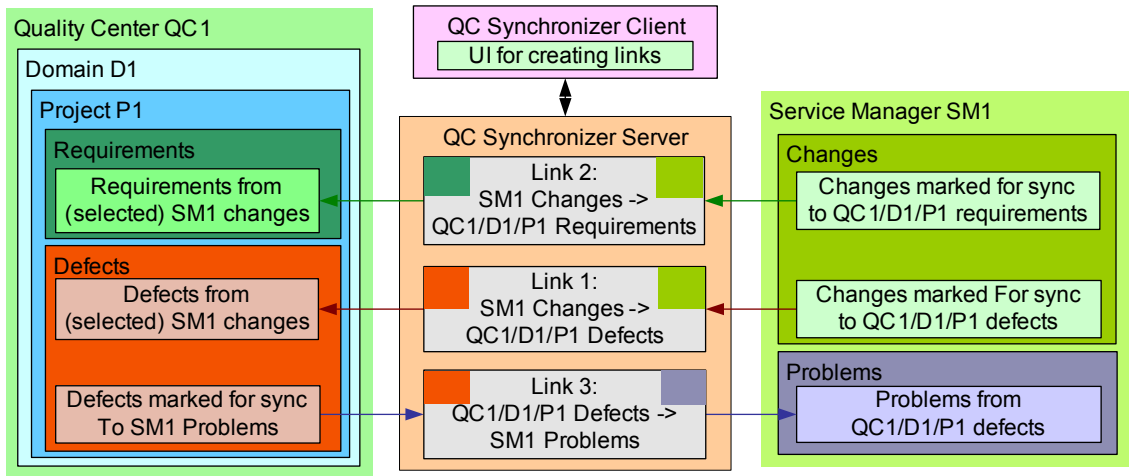
```
<itg:field name="Urgency" type="Single_Value_List" readonly="false"
          required="mandatory">
  <itg:items>
    <itg:item value="1">1 - Critical</itg:item>
    <itg:item value="2">2 - High</itg:item>
    <itg:item value="3">3 - Average</itg:item>
    <itg:item value="4">4 - Low</itg:item>
  </itg:items>
</itg:field>
</itg:module>
</itg:mapping>
```


6 Link Configuration

This chapter shows how to configure and test links.

- *Link Configuration Basics* on page 48
- *SM Change -> QC Defect* on page 55
- *SM Change -> QC Requirement* on page 61
- *SM Problem to QC Defect* on page 65

The following diagram summarizes link configuration.



Link Configuration Basics

This section describes aspects of link creation that are common to all three types of links.

- *Create a Link*
- *QC Field <-> SM Field mappings*
- *List Value Mappings*
- *Constant -> SM field mappings*

Filters are only required for SM Change -> QC Requirement (see *Filters* on page 65). The events settings determine what QCS does in response to specified events. Events must be specified for all three link types.

Create a Link

The following table summarizes the properties required in the wizard. Have this data available before starting the wizard.



A link can not be duplicated. For example, if a link already exists for SMServer1/Changes -> QCServer1/Doman1/Project1/Defects, then a second link between these 2 entities can not be created.

Table 5 Wizard Link Properties

End point	Parameter	Constraint
QC	Username	
QC	Password	
QC	Server URL	
QC	Domain	
QC	Project	
SM	User name	Required
SM	Password	
SM	Service URL	Required http://<service_manager_host>:<port>/sc62server/ PWS/QCIntChangeService.wsdl or http://<service_manager_host>:<port>/sc62server/ PWS/QCIntProblemService.wsdl
SM	Adapter Configuration (SM field values) filename	Empty or the adapter data folder file (see <i>Copy SM Adapter Configuration Files</i> on page 42).
SM	QCProject	Required (because of an adapter limitation). Format is qc_host/qc_domain/qc_project

To create a link:

- 1 Click **Link / Create**. The “Step 1: Assign general properties” dialog appears.
- 2 Enter the required information (the following example is for SM Change -> QC Defect).

The screenshot shows a dialog box titled "Create Link - Step 1 of 4 - General Properties". On the left is the HP Quality Center Synchronizer logo. The main area is titled "Assign general properties:" and contains the following fields:

- Link name:
- Description:
- Endpoint 1 type:
- Endpoint 2 type:

- 3 Click **Next**. The “Step 2: Assign QC endpoint connection properties” dialog appears.
- 4 Enter the required information (see table below for details).

The screenshot shows a dialog box titled "Create Link - Step 2 of 4 - Quality Center Endpoint". On the left is the HP Quality Center Synchronizer logo. The main area is titled "Assign Quality Center endpoint connection properties:" and contains the following fields:

- User name:
- Password:

Parameter	Value
ServerURL	http://localhost:8080/qcbin
Domain	DEFAULT
Project	Test44

At the bottom right is a "Check Connectivity" button.

- 5 Click **Next**. One of the following appears:
 - “Step 3: Assign SM ChangeManagement endpoint connection properties”
 - “Step 3: Assign SM ProblemManagement endpoint connection properties”
- 6 Enter the required information (the following example is for SM Change -> QC Defect).

The screenshot shows a dialog box titled "Create Link - Step 3 of 4 - SM ChangeManagement Endpoint". On the left is the HP Quality Center Synchronizer logo. The main area is titled "Assign SM ChangeManagement endpoint connection properties:" and contains the following fields:

- User name:
- Password:

Parameter	Value
Configuration File Name	configuration_file_default.xml
QC Project	localhost/DEFAULT/Test44
Change Service URL	http://localhost:13080/sc62server/PWS/QCIntChangeService.wsdl

At the bottom right is a "Check Connectivity" button.



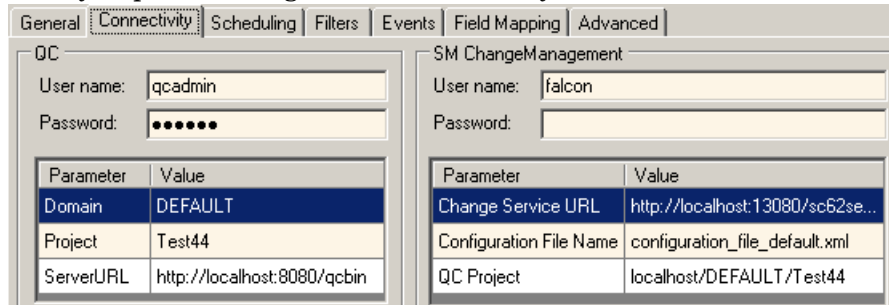
fQC Project has the same value as specified on SM customization.

- 7 Click **Next**. If this is a change management link, then “Step 4: Select entity types” dialog appears.
- 8 Select one of the following:
 - **Change as Defect**

Change as Requirement



- 9 Click **Save**. The link is created.
- 10 Modify required settings in the Connectivity tab.

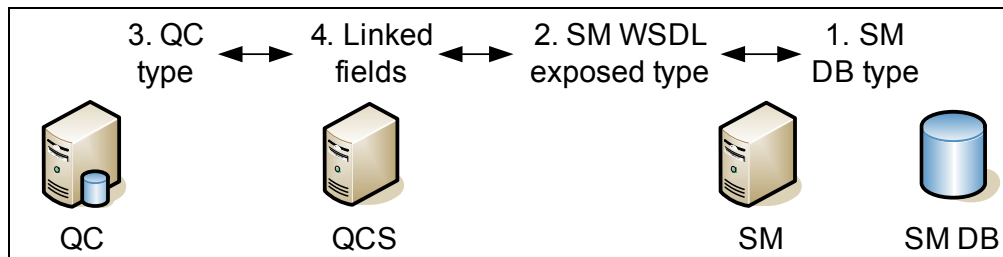


QC Field <-> SM Field mappings

This section describes how to map QC fields and SM fields.

- *Create Mapping*
- *General Mapping Requirements*
- *Matching Types*

The following diagram shows the field mapping chain.



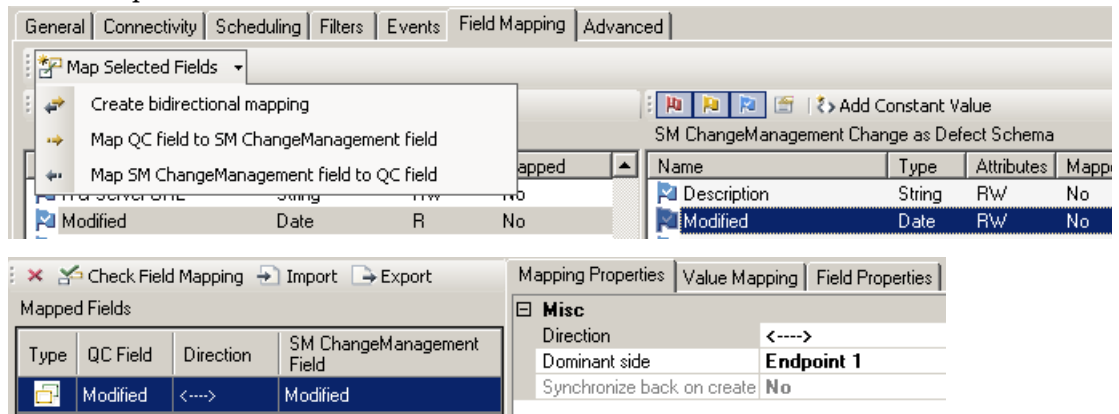
For examples of field mappings see

- *SM Change -> QC Defect , QC Field <-> SM Field on page 56)*
- *SM Change -> QC Requirement, QC Field <-> SM Field on page 62)*
- *SM Problem to QC Defect, QC Field <-> SM Field on page 66)*

Create Mapping

To map fields, follow these steps.

- 1 Select a field on each side.
- 2 From “Map Selected Fields” select the direction.



General Mapping Requirements

When creating field links, keep the following limitations in mind:

- If you change the mappings you must do a full synchronization to ensure synchronization of historical data. Otherwise, your historical data can not be synchronized correctly and you may get errors in the next incremental synchronization.
- A field in one endpoint can be mapped to only one field in the other endpoint.
- Mandatory fields must be mapped. If a null value is written to a mandatory field, an error will occur at runtime.
- If you map string fields with different maximum lengths, during synchronization a string value in the source endpoint will be truncated as necessary if it exceeds the maximum length of the corresponding field.

Matching Types

The following table shows the allowed data type combinations. Highlighted entries are demonstrated in examples in this document.

Table 6 Data Type Combinations

QC Data type	QCS QC Type	Dir	QCS SM type	WSDL data type (SM exposed data type)	SM DB data type
Number	Number	<->	Number	DecimalType or IntType (8)	Number (7)
String	String	<->	String	(9)	Character
Lookup List	Single value list	<->	Single value list (5)	Character (default) / StringType	Character
Date (6)	Date	<->	Date	DateTimeType (required) (1)	Date/time
Lookup List	Multi-value List	<->	Multi-value List	StringType (3)	Array (4)
String	String	<->	String	BooleanType	Logical
Memo	String	<->	String	StringType	Character
User List (2)	User List	->	String	StringType (10)	Character

Notes:

- All types except 'Date' type do not need to explicitly specify type on WSDL configuration. For details see “Best Practices for Publishing and Consuming Web Services with ServiceCenter”.
- (2) Write to the QC field "User_List" only if SM has exactly the same users (including logins, names, etc.) as QC. An incorrect entry can cause serious problems in QC. You can read from QC "User_List" field and write to SM "String" type field only if the field in SM is NOT a field with SM logins.
- (3) Only the character type of array is supported
- (4) Only an array of characters is supported.
- (5) List values are from an XML file which can only contain strings.

- (6) QC data only supports Yr/Mo/Dt.
- (8) IntType supports a data range from -2,147,483,648 to 2,147,483,647.
- (9) It is recommended to leave this field blank. Otherwise "Invalid byte 2 of 3-byte UTF-8 sequence" might occur if certain I18N characters are synchronized.
- (10) Supports 2-way synchronization, but the names must be the same.

The following table shows the document sections where types are specified.

QC DB	SM WSDL	SM DB
	<i>Change Management / Specify Field External Access on page 20</i>	
	<i>Problem Management / Specify Field External Access on page 24</i>	
		<i>Change Management / Add Fields on page 19</i>
		<i>Problem Management / Add Fields on page 24</i>
<i>Defects / Add Fields on page 31</i>		
<i>Requirements / Add Fields on page 35</i>		

List Value Mappings

This section describes how to map values for multi-valued lists.



QCS does not have access to the values of SM multi-valued lists, and therefore the values must be specified in an XML file.

For examples of list field mapping see

- *SM Change -> QC Defect* (section *QC List Values <-> SM List Values (XML)* on page 57)
- *SM Problem to QC Defect* (section *QC List Values <-> SM List Values (XML)* on page 67)

Some multi-value fields also require mapping of available values (as shown in the following diagram).

Name	Type	Attributes	Mapped	Name	Type	Attributes	Mapped
Severity	Single value list	RW	No	Urgency	String	RW	No

Mapped Fields		Mapping properties	Value mapping	Field properties
Severity	<---->	Urgency		
Endpoint 1 value	Direction	Endpoint 2 value		
5-Urgent	<---->	1 - Critical		
4-Very High	<---->	2 - High		
3-High	<---->	3 - Average		

Constant -> SM field mappings

For examples of constant -> SM field mappings see

- *SM Problem to QC Defect* (section *Constants -> SM List Value Fields* on page 69)

The following diagram shows an example of constant -> SM field mapping.

Type	Endpoint 1 Field	Direction	Endpoint 2 Field
Value: AUTO		---->	AssignmentGroup
Value: client system		---->	Category
Value: BOB.HELPDESK		---->	ProblemOwner

SM Change -> QC Defect

This section describes how to create and test a link.

- *Specify Endpoints / Type of Link*
- *Field Mappings*
- *Events*
- *Test*

Specify Endpoints / Type of Link

Specify the connection properties as described in *Create a Link* on page 48 with the following settings specific for this type of link:

- 1 Step 1 endpoint 2 type = **SM ChangeManagement**.
- 2 Step 3 service URL =
http://<service_manager_host>:<port>/sc62server/PWS/QCIntChangeService.wsdl
- 3 Step 4 select entity types = **Change as Defect**.

Field Mappings

To configure the field mappings, follow these steps.

- *QC Field <-> SM Field*
- *QC List Values <-> SM List Values (XML)*

QC Field <-> SM Field

Specify mappings as shown in following figure (the settings are summarized in the table below).

Type	Endpoint 1 Field	Direction	Endpoint 2 Field
	Change id	<---	ChangeID
	Defect ID	--->	QCEntityID
	Summary	<--->	Description
	Severity	<--->	Urgency

The following table summarizes the field mappings.

Table 7 SM Change -> QC Defect Mappings

QC Len	QC DB Name(3)	QC Type	QC/QCS Label	QCS QC Type	Dir	QCS SM type	QCS Name / SM WSDL Caption	SM WSDL Type	SM cm3r DB Name / SM WSDL field	cm3r DB type	SM Len (4)
40	BG_USER_02	Str	Change id	String	<-	String	Change ID	StringType	header,number	Char	100
10	BG_BUG_ID	Num	Defect ID (1)	Number	->	Number	QCEntityID (2)	IntType	qcintegration.id	Num	xx
255	BG_SUMMARY	Str	Summary	String	<->	String	Description	StringType	description.structure,description	Char	xx
70	BG_SEVERITY	Lookup List	Severity	Single val list	<->	Single val list	Urgency	Character	severity	Char	40

Notes

1. Only the highlighted rows are mandatory.
2. QC defect ID is assigned in QC only after saving a new defect.
3. Check for QCEntityID mapping property "Synchronize back on create".
4. The numeric suffix on generated names may differ on your system.
5. The length may differ on your system.

The following table shows where the mappings were specified in previous sections.

Table 8 SM Change -> QC Defect Mapping Sections

			(this section)								
							<i>Change Management / Specify Field External Access</i> on page 20				
			<i>Defects / Add Fields</i> on page 31						<i>Change Management / Add Fields</i> on page 19		
QC Len	QC DB Name	QC Type	QC/QCS Label	QCS QC Type	Dir	QCS SM type	QCS Name/SM WSDL Caption	SM WSDL Type	cm3r DB Name	cm3r DB type	SM Len

QC List Values <-> SM List Values (XML)

Specify mappings as shown in following figure (the settings are summarized in the table below).

Mapped Fields				Mapping properties	Value mapping	Field properties
Type	Endpoint 1 Field	Direction	Endpoint 2 Field	Endpoint 1 value	Direction	Endpoint 2 value
	Change id	<--->	ChangeID	5-Urgent	<--->	1 - Critical
	Defect ID	--->	QCEntityID	4-Very High	<--->	2 - High
	Summary	<--->	Description	3-High	<--->	3 - Average
	Severity	<--->	Urgency	2-Medium	<--->	4 - Low
				1-Low	<--->	(Empty string)

The following table summarizes the mappings you just created between the single value lists (which have their own direction).

Table 9 SM Change -> QC Defect List Value Mappings

QC Len	QC DB Name	QC Type / Lookup list values	QC/QCS Label	QCS QC Type	Dir			QCS SM type	QCS Name/ SM WSDL Caption	SM WSDL Type	cm3r DB Name // SM WSDL Field	cm3r DB type	SM Len
					QC value (from lookup list) (1)	value map dir	SM field value (from SM Adapter xml config file for SM Change Management) (2)						
70	BG_SEVERITY	Lookup List	Severity	Single value list	<->			Single value list	Impact	String Type	severity	Char	40
					5-Urgent	<->	1-Critical						
					4-VeryHigh	<->	2-High						
					3-High	<->	3-Average						
					2-Medium	<->	4-Low						
					1-Low	<->	(Empty string)						



Notes:

1. Lookup list is created in QC.

Field Settings

Field Name: BG_SEVERITY

Field Label: Severity

Field Type: Lookup List

Lookup List

Severity [v] [New List] [Goto List]

Verify Value

Project Lists

Lists: Severity [v]

List Items

- 1-Low
- 2-Medium
- 3-High
- 4-Very High
- 5-Urgent

2. The XML file is in C:\Program Files\HP\Quality Center Synchronizer\adapters\dat\SM ChangeManagement\configuration_file_default.xml (partial listing).

```
<?xml version="1.0" encoding="UTF-8"?>
<itg:mapping xmlns:itg="http://www.hp.com/smci/SMQCIntegration/config">
  <itg:module name="change">
    <itg:field name="Urgency" tye="Single_Value_List" readonly="false"
      required="mandatory" length="50">
      <itg:items>
        <itg:item value="1">1 - Critical</itg:item>
```

```

    <itg:item value="2">2 - High</itg:item>
    <itg:item value="3">3 - Average</itg:item>
    <itg:item value="4">3 - Low</itg:item>
  </itg:items>
</itg:field>
</itg:module>

```

The following table shows where the mappings were specified in previous sections.

Table 10 QC Defect -> SM Problem value list mappings specified

(this section)													
<i>Defects / Add Fields</i> on page 31				<i>Copy SM Adapter Configuration Files / SM Change Management</i> on page 43				<i>Change Management / Specify Field External Access</i> on page 20				<i>Change Management / Add Fields</i> on page 19	
QC Len	QC DB Name	QC Type/ Lookup list values (defined in QC)	QC/ QCS Label	QCS QC Type	Dir			QCS SM type	QCS Name/ SM WSDL Caption	SM WSDL Type	cm3r DB Name / SM WSDL Field	cm3r DB type	SM Len
					Link QC list field / selected value -> SM list field / selected string (from xml)								
					QC val (from lookup list) (1)	Value map Dir	SM field value (from SM Adapter xml config file for SM Problem Management) (2)						

Events

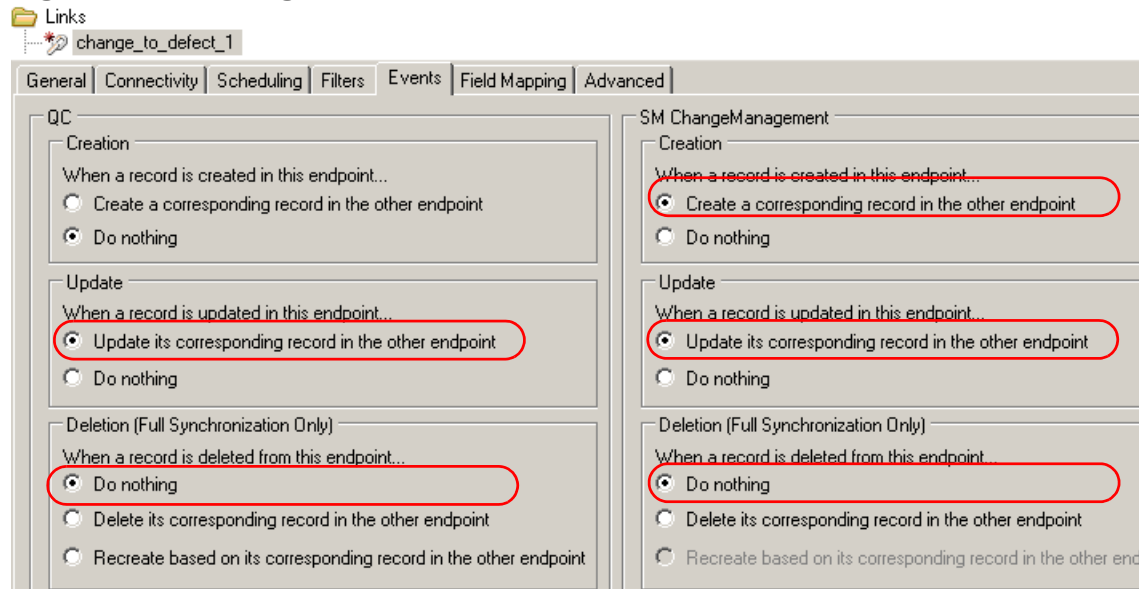
The following table lists the events.

Table 11 SM Change -> QC Defect Events

Events Tab Settings	SM Action (Event)	QC Action (Event)	QCS Action
SM / Create Record	Create SM Change (Forward -> QC Defect)		Create QC Req
	Create SM Change (NO forward)		(none)
SM / Update	Update		Update
QC / Update		Update	Update
SM / Do nothing	Delete		(none)
QC / Do nothing		Delete	(none)

The following diagram shows the settings.

Figure 2 SM Change -> QC Defect Events



Test

This section shows how to test the link.



A mandatory field (in either SM and QC) does not accept a null value. Synchronization may fail if a mandatory field is mapped to a field that is null.

The following is just an example. The exact steps required on your system may differ significantly. The phase in which the tab for QC Integration appears may be different on your system.

- 1 Save the configuration (an integrity check is automatically run).
- 2 Click **Enable Link**.
- 3 Create a Service Manager change (the category of the change depends on each Service Manager customizations; RFC is used in this example).

Category	Description
RFC	Request For Change

Request For Change

General | **Description**

Description of Change

Change to Defect

- 4 Change the phase to **building**. The “QC Integration” tab appears.

Phase Name	Description
Assessment	Assessment
Building	RFC Building
RFC Implementation	Place Changes Into Service
RFC Testing	RFC Testing

- 5 Select **Forward as Requirement**.

Request For Change

qc integration | General | Description | Associated

Forward to QC: 2 - Forward as Defect

Defect ID

server/domain/project localhost/DEFAULT/Test44

- 6 Synchronize.

Cancel Current Task | View Report | Refresh Progress | Auto Refresh

Running: Connecting to endpoint 1...
Running: Connecting to endpoint 2...
Running: Querying non filtered set
Running: Handling endpoint 1 - Processing entity #1 of #1 in the Create list, (Total: passed = 0, failed = 0)
Passed: Disconnecting...
Completed : Passed

- 7 View the defect in QC.

Defects | Edit | View | Favorites | Analysis

Defect ID	Status	Assigned To	Priority	Summary
10	New			Change to Defect

Details | SM Integration (Details)

Forward as problem: Change to Defect | Change id: C34

Problem ID:

SM Change -> QC Requirement

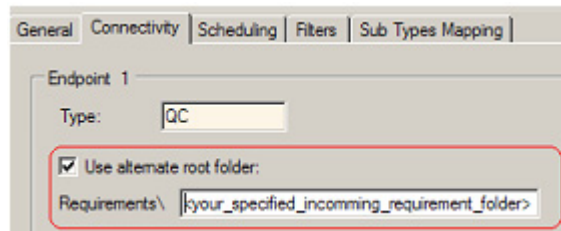
This section describes how to create and test a link.

- *Specify Endpoints / Type of Link*
- *Field mappings*
- *Events*
- *Test*

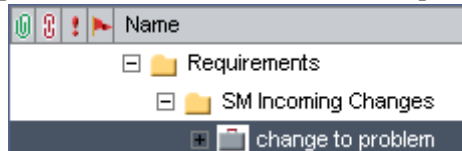
Specify Endpoints / Type of Link

Specify the connection properties as described in *Create a Link* on page 48 with the following settings specific for this type of link:

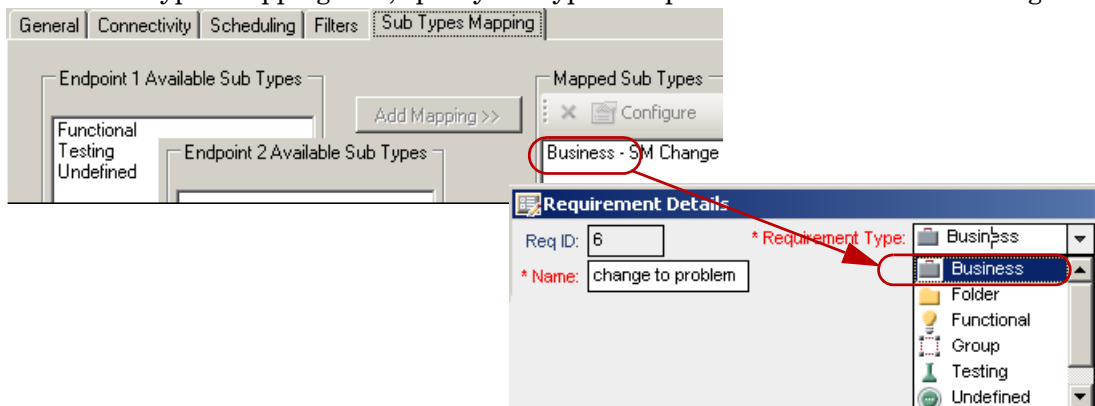
- 1 Step 1 endpoint 2 type = **SM ChangeManagement**.
- 2 Step 3 service URL =
http://<service_manager_host>:<port>/sc62server/PWS/QCIntChangeService.wsdl
- 3 Step 4 select entity types = **Change as Requirement**.
- 4 Specify the incoming requirement folder as shown in the following figure.



Requirements will be created in the specified folder in QC.



- 5 In the "sub types mapping" tab, specify the type of requirements created from changes.



Field mappings

Configure the following types of field mappings.

- *QC Field <-> SM Field*

QC Field <-> SM Field

Specify mappings as shown in following figure (the settings are summarized in the following table).

Mapped Fields			
Type	Endpoint 1 Field	Direction	Endpoint 2 Field
	Change ID in SM	<---	ChangeID
	Name	<---	Description
	Req ID	--->	QCEntityID

The following table summarizes the field mappings.

Table 12 SM Change -> QC Requirement Mappings

QC Len	QC DB Name	QC Type	QC/QCS Label	QCS QC Type	Dir	QCS SM type	QCS Name / SM WSDL Caption	SM WSDL Type	cm3r DB Name	cm3r DB type	SM Len
40	RQ_USER_01	Str	Change ID in SM	String	<-	String	Change ID	StringType	header,number	Char	100
10	RQ_REQ_ID	Num	Req ID (1)	Number	->	Number	QCEntityID	IntType	qcintegration.id	Num	xx
255	RQ_REQ_NAME	Str	Name	String	<->	String	Description	--	description.structure,description	Char	xx

Notes:

1. Only the highlighted rows are mandatory.
2. Check for QCEntityID mapping property "Synchronize back on create".

The following table shows where the mappings were specified in previous sections.

Table 13 SM Change -> QC Requirements Mapping Sections

Link QC field <-> SM field											
								Change Management / Specify Field External Access on page 20			
Requirements / Add Fields on page 35								Change Management / Add Fields on page 19			
QC Len	QC DB Name	QC Type	QC/QCS Label	QCS QC Type	Dir	QCS SM type	QCS Name/SM WSDL Caption	SM WSDL Type	cm3r DB Name	cm3r DB type	SM Len

Events

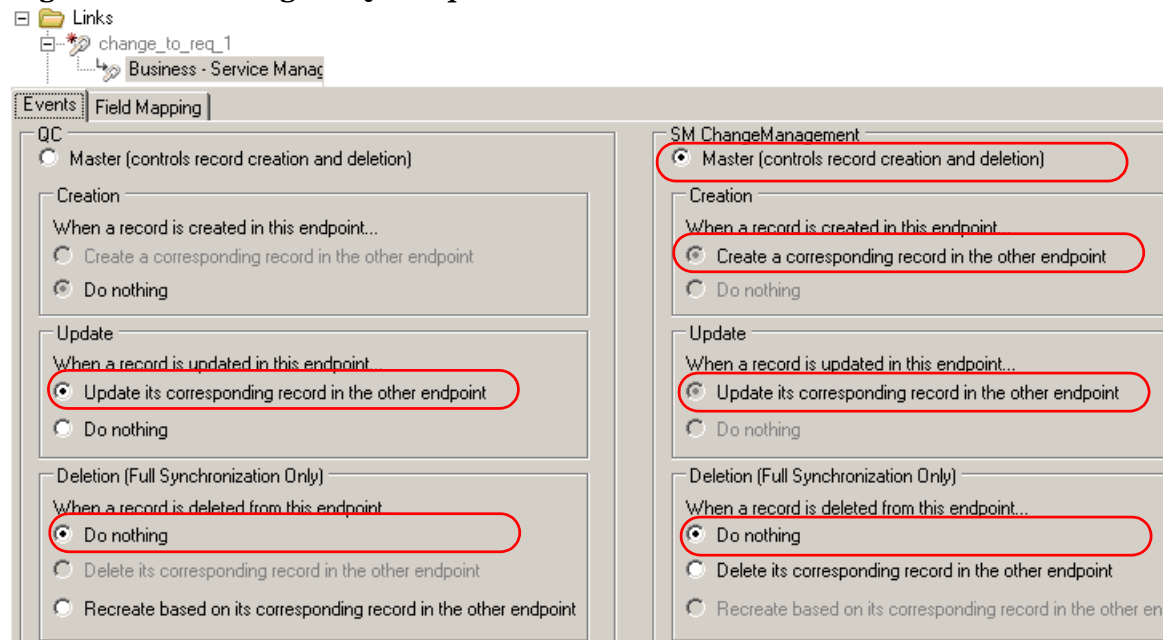
The following table lists the events.

Table 14 SM change -> QC Requirements Events

Events Tab Settings	SM Action (Event)	QC Action (Event)	QCS Action
SM / Master			
(SM / Create record)	Create SM Change (forward -> QC Req)		Create QC Req
	Create SM Change (NO forward)		(none)
(SM / Update)	Update		Update
QC / Update		Update	Update
SM / Do nothing	Delete		(none)
QC / Do nothing		Delete	(none)

The following diagram shows the settings.

Figure 3 SM Change -> QC Requirements Events



Test

This section shows how to test the link.



A mandatory field (in either SM and QC) does not accept a null value. Synchronization may fail if a mandatory field is mapped to a field that can be null.

The following is just an example. The exact steps required on your system may differ significantly. The phase in which the tab for QC Integration appears may be different on your system.

- 1 Save the configuration (an integrity check is automatically run).
- 2 Click **Enable Link**.
- 3 Create a Service Manager change (the category of the change depends on each Service Manager customizations; RFC is used in this example).

Changes	Category	Description
Open New Change	RFC	Request For Change

Request For Change

General | **Description**

Description of Change

Change to Requirement

- 4 Change the phase to building. The “QC Integration tab” appears.

Phase Name	Description
Assessment	Assessment
Building	RFC Building
RFC Implementation	Place Changes Into Service
RFC Testing	RFC Testing

- 5 Select **Forward as Requirement**.

qc integration | General | Description | **Associa**

Forward to QC: 1 - Forward as Requirement

server/domain/project localhost/DEFAULT/Test44

- 6 Synchronize.

Cancel Current Task | View Report | Refresh Progress | Auto Refresh

Running: Connecting to endpoint 1...
Running: Connecting to endpoint 2...
Running: Querying non filtered set
Running: Handling endpoint 1 - Processing entity #1 of #1 in the Create list, (Total: passed = 0, failed = 0)
Passed: Disconnecting...
Completed : Passed

- 7 View the requirement in QC.

Requirements | Edit | View | Favorites | Analysis

Name	Direct Cover Status	Req ID	Author
Requirements	----	0	
SM Incoming Changes	----	5	qcadmin
Change to Requirement	----	6	

Requirement Details

Req ID: 6 * Name: Change to Requirement

Details | Page 2 * Requirement Type: Business

Change ID in SM: C35

SM Problem to QC Defect

This section describes how to create and test a link.

- *Specify Endpoints / Type of Link*
- *Filters*
- *Field Mappings*
- *Events*
- *Test*

Specify Endpoints / Type of Link

Specify the connection properties as described in *Create a Link* on page 48 with the following settings specific for this type of link:

- 1 Step 1 endpoint 2 type = **SM ProblemManagement**.
- 2 Step 3 “Configuration File Name” to `configuration_file_default.xml`
- 3 Step 3 service URL (required) to `http://<service_manager_host>:<port>/sc62server/PWS/QCIntProblemService.wsdl`
- 4 Step 4 select entity types = **Problem by Defect**.

Filters

In the Filters tab, select filter **SMIntegrationView** for endpoint 1 (QC). If the filter is not available, see *Create a View* on page 33.

The screenshot shows a configuration window with several tabs: General, Connectivity, Scheduling, Filters, Events, and Field Mapping. The 'Filters' tab is active. It contains two sections: 'Endpoint 1 Filters' and 'Endpoint 2 Filters'. In the 'Endpoint 1 Filters' section, the 'No Filter' radio button is unselected, and the 'Use filter (for creation events):' radio button is selected. Below this, a dropdown menu is open, showing 'Public: SMIntegrationView'. In the 'Endpoint 2 Filters' section, the 'No Filter' radio button is selected, and the 'Use filter (for creation events):' radio button is unselected. Below this, a dropdown menu is also present but empty.

Field Mappings

To configure the field mappings, follow these steps.

- *QC Field <-> SM Field*
- *QC List Values <-> SM List Values (XML)*
- *Cosntants -> SM List Value Fields*

QC Field <-> SM Field

Specify mappings as shown in following figure (the settings are summarized in the table below).

Type	Endpoint 1 Field	Direction	Endpoint 2 Field
	Summary	<---->	Description
	Severity	<---->	Impact
	Problem ID	<---	ProblemID
	Defect ID	---->	QCEntityID

The following table summarizes the field mappings.

Table 15 QC Defect -> SM Problem Mappings

QC Len	QC DB Name	QC Type	QC/QCS Label	QCS QC Type	Dir	QCS SM type	QCS Name/ SM WSDL Caption	SM WSDL Type	rootcause DB Name / SM WSDL Field	rootca use DB type	SM Len
40	BG_USER_03	String	Problem id	String	<-	String	Problem ID	StringType	id	Char	100
10	BG_BUG_ID	Number	Defect ID	Number	->	Number	QCEntityID	IntType	qcintegration.id	Num	xx
255	BG_SUMMARY	String	Summary	String	<->	String	Description	--	description	Char	xx
70	BG_SEVERITY	Lookup List	Severity	Single value list	<->	Single value list	Impact	StringType	initial.impact	Char	40

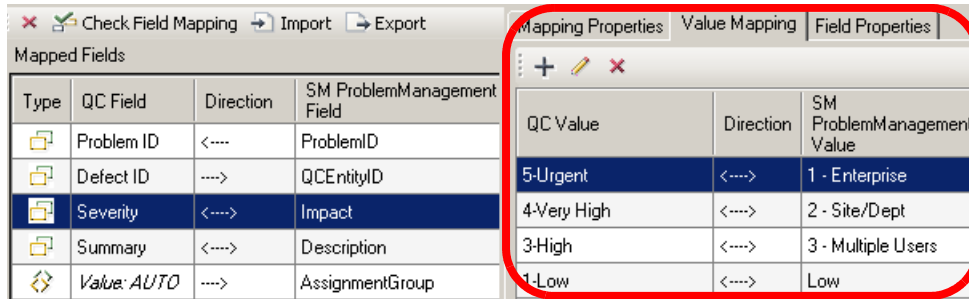
The following table shows where the mappings were specified in previous sections.

Table 16 QC Defect -> SM Problem Mapping Sections

			(this section)								
							<i>Problem Management / Specify Field External Access on page 24</i>				
			<i>Defects / Add Fields on page 31</i>						<i>Problem Management / Add Fields on page 24</i>		
QC Len	QC DB Name	QC Type	QC/QCS Label	QCS QC Type	Dir	QCS SM type	QCS Name/ SM WSDL Caption	SM WSDL Type	rootcause DB Name / SM WSDL Field	rootcause DB type	SM Len

QC List Values <-> SM List Values (XML)

Specify mappings as shown in following figure (the settings are summarized in the following table).



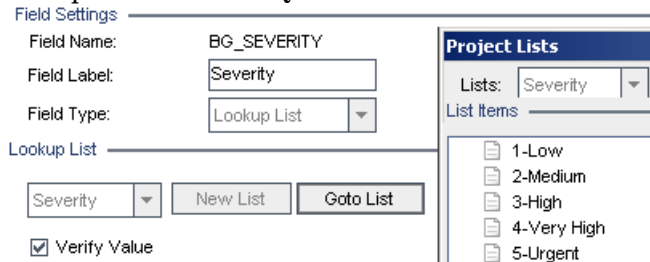
The following table summarizes the mappings you just created between the single value lists (which have their own direction).

Table 17 QC Defect -> SM Problem List Value Mappings

QC Len	QC DB Name	QC Type/ Lookup list value (defined in QC)	QC/ QCS Label	QCS QC Type	Dir			QCS SM type	QCS Name/ SM WSDL Caption	SM WSDL Type	rootcaus e DB Name / SM WSDL Field	rootca use DB type	SM Len
					QC value (from lookup list) (1)	value map Dir	SM field value (from SM Adapter xml file for SM Problem) (2)						
70	BG_SEVERITY	Lookup List	Severity	Single value list	<->			Single value list	Impact	String Type	initial.impact	Character	40
					5-Urgent	<->	1-Enterprise						
					4-VeryHigh	<->	2-Site/Dept						
					3-High	<->	3-Multiple Users						
					2-Medium	<->	4-User						
					1-Low	<->	"Low" (constant)						

Notes:

1. Lookup list created in QC.



2. The XML file in C:\Program Files\HP\Quality Center Synchronizer\adapters\dat\SM ProblemManagement\configuration_file_default.xml (partial listing).

```
<?xml version="1.0" encoding="UTF-8"?>
<itg:mapping xmlns:itg="http://www.hp.com/smci/SMQCIntegration/config">
<itg:module name="change">
```

```

<itg:field name="Urgency" tye="Single_Value_List" readonly="false"
required="mandatory" length="50">
  <itg:items>
    <itg:item value="1">1 - Critical</itg:item>
    <itg:item value="2">2 - High</itg:item>
    <itg:item value="3">3 - Average</itg:item>
    <itg:item value="4">4 - Low</itg:item>
  </itg:items>
</itg:field>
</itg:module>

```

The following table shows where the mappings were specified in previous sections.

Table 18 QC Defect -> SM Problem List Value Mapping Sections

(this section)													
<i>Defects / Add Fields</i> on page 31				<i>Copy SM Adapter Configuration Files / SM Problem Management</i> on page 44				<i>Problem Management / Specify Field External Access</i> on page 24				<i>Problem Management / Add Fields</i> on page 24	
QC Len	QC DB Name	QC Type / Lookup list values (defined in QC)	QC/QCS Label	QCS QC Type	Dir			QCS SM type	QCS Name/ SM WSDL Caption	SM WSDL Type	rootcause DB Name / SM WSDL Field	rootca use DB type	SM Len
					Link QC list field / selected value -> SM list field / selected string (from xml)								
					QC value (from look up list) (1)	value map Dir	SM field value (from SM Adapter xml file for SM Problem) (2)						

Cosntants -> SM List Value Fields

Specify mappings as shown in following figure (the settings are summarized in the table below).

3_defect_to_problem

Mapped Fields			
Type	Endpoint 1 Field	Direction	Endpoint 2 Field
↔	Value: <i>AUTO</i>	---->	AssignmentGroup
↔	Value: <i>client system</i>	---->	Category
↔	Value: <i>BOB.HELPDESK</i>	---->	ProblemOwner
↔	Value: <i>exchange</i>	---->	ProblemType
↔	Value: <i>email client</i>	---->	ProductType
↔	Value: <i>localhost/DEFAULT/T...</i>	---->	QCProject
↔	Value: <i>2 - High</i>	---->	Severity
↔	Value: <i>software</i>	---->	SubCategory
↔	Value: <i>ITIL</i>	---->	WorkFlowType

The following table summarizes the mappings.

Table 19 Constant Values -> SM Problem List Values Mapping

QCS SM constant value	QCS Name/SM WSDL Caption	SM WSDL Type	rootcaus e DB type	rootcause DB Name / SM WSDL Field	SM Len
(your setup) (1)	QCProject	StringType	Character	qcintegration.project	60
	Modified	DateTimeType	Date/time	sysmodtime	xx
AUTO	AssignmentGroup	StringType	Character	assignment	50
client system	Category (2)	StringType	Character	incident.category	40
BOB.HELPDESK	ProblemOwner	StringType	Character	ticket.owner	40
exchange	ProblemType	StringType	Character	problem.type	40
email client	ProductType	StringType	Character	product.type	40
2 - High	Severity	StringType	Character	severity	50
software	SubCategory	StringType	Character	subcategory	40
ITIL	WorkFlowType	StringType	Character	category	40



1. Required.
2. Add constant mappings for Endpoint 2 (Problem) categories. By default, an SM problem has 4 level categories: Category/SubCategory/ProductType/ProblemType. Select the category for incoming problems. Note that the category hierarchy can be customized by the SM administrator (in SM IIA you don't have to specify these values). Ask the SM administrator about the problem category and expose problems correctly via SM SOAP interface.

The following table shows where the mappings were specified in previous sections.

Table 20 Constant -> SM Problem List Values Mapping Sections

Link selected string (from xml) -> SM single value list field					
Copy SM Adapter Configuration Files / SM Problem Management on page 44	Problem Management / Specify Field External Access on page 24				
			Problem Management / Add Fields on page 24		
Single value lists of strings (in SM Adapter config file for SM Problem Management) (1)	QCS Name/ SM WSDL Caption	SM WSDL Type	rootcause DB Name / SM WSDL Field	rootcause DB type	SM Len



Notes:

1. xml file in C:\Program Files\HP\Quality Center Synchronizer\adapters\dat\SM ProblemManagement\configuration_file_default.xml (partial listing).

```
<?xml version="1.0" encoding="UTF-8"?>
<itg:mapping xmlns:itg="http://www.hp.com/smci/SMQCIntegration/config">
  <itg:module name="problem">
    <itg:field name="AssignmentGroup" type="Single_Value_List"
      required="mandatory">
      <itg:items>
        <itg:item value="AUTO">AUTO</itg:item>
        <itg:item value="DEFAULT">DEFAULT</itg:item>
        <itg:item value="FIRSTLINE">FIRSTLINE</itg:item>
        <itg:item value="SECONDLINE">SECONDLINE</itg:item>
        <itg:item value="HELPDESK">HELPDESK</itg:item>
      </itg:items>
    </itg:field>
    <itg:field name="Modified" type="Date" readonly="true" />
  </itg:module>
</itg:mapping>
```

Events

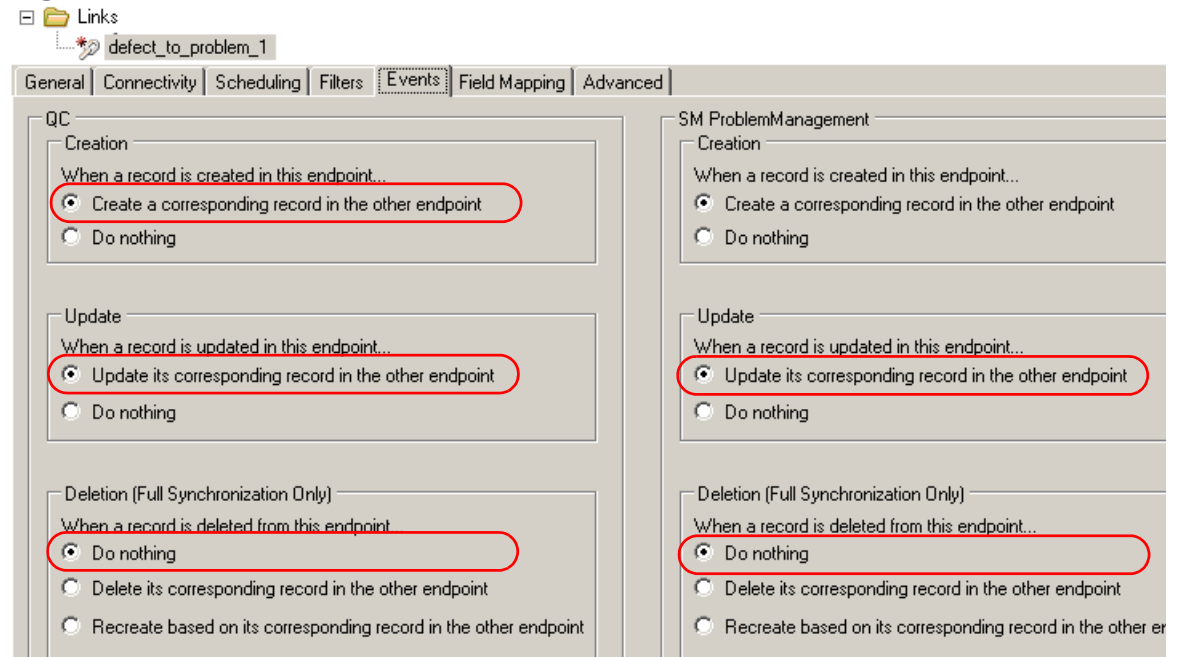
The following table lists the events.

Table 21 QC Defect -> SM Problem Event Settings

Events Tab Settings	QC Action (Event)	SM Action (Event)	QCS Action
QC / Create record	Create QC Defect (forward -> SM Problem)		Create SM problem
	Create QC Defect (NO forward)		(none)
QC / Update	Update		Update
SM / Update		Update	Update
QC / Do nothing	Delete		(none)
SM / Do nothing		Delete	(none)

The following diagram shows the settings.

Figure 4 QC Defect -> SM Problem Events



Test

This section shows how to test the link.



A mandatory field (in either SM and QC) does not accept a null value. Synchronization may fail if a mandatory field is mapped to a field that can be null.

The following is just an example. The exact steps required on your system may differ significantly. The phase in which the tab for QC Integration appears may be different on your system.

- 1 Save the configuration (an integrity check is automatically run).
- 2 Click **Enable Link**.
- 3 Create a QC defect.
- 4 **Select Forward as problem.**

The screenshot shows a 'Defect Details' form. At the top, there is a table with columns: Defect ID, Status, Assigned To, and Summary. The first row contains: 13, New, (blank), and Defect_to_Problem_1. Below the table, the 'Defect' field contains '13' and the 'Summary' field contains 'Defect to Problem 1'. Under the 'Details' section, there is a tab labeled 'SM Integration (Details)'. In this tab, the 'Forward as problem:' dropdown menu is set to 'Y'. There is also a 'Change id:' field and a 'Problem ID:' field.

- 5 Synchronize.

The screenshot shows a synchronization progress window. At the top, there are buttons: 'Cancel Current Task', 'View Report', 'Refresh Progress', and 'Auto Refresh' (checked). Below the buttons, the progress status is shown as follows:
Running: Connecting to endpoint 1...
Running: Connecting to endpoint 2...
Running: Querying non filtered set
Running: Handling endpoint 1 - Processing entity #1 of #1 in the Create list. (Total: passed = 0, failed = 0)
Passed: Disconnecting...
Completed : Passed

- 6 View the problem in SM.

The screenshot shows the 'Problem Control - Problem Identification and Classification' form. It includes the following fields:
Record Number: PM0162
Brief Description: defect to problem
Below these fields are tabs for 'QC Integration', 'Classification', 'Activities', and 'Attachments'. The 'QC Integration' tab is selected. It contains:
Defect ID: 14
server/domain/project: localhost/DEFAULT/Test44

- 7 View the problem in QC.

The screenshot shows a 'Defect Details' form. The 'Defect' field contains '14' and the 'Summary' field contains 'defect to problem terry 2'. Under the 'Details' section, there is a tab labeled 'SM Integration (Details)'. In this tab, the 'Forward as problem:' dropdown menu is set to 'Y'. There is also a 'Change id:' field and a 'Problem ID:' field containing 'PM0162'.

A Known Issues

This appendix describes the following known issues:

- *Timestamp*
- *SM/QC Locked Record*
- *Moving a Synchronized Requirement Stop Synchronization*
- *QC 4-byte Integer Limitation*
- *QC Date Field*
- *QC Special Characters*

Timestamp

When an error occurs, reset the timestamp to avoid a full synchronization. Reset the time stamp for both the link and link mappings to any time before the error occurred.

SM/QC Locked Record

Synchronization could fail if SM or QC locks the record that is being updated by users from SM or QC client. So the record changes failed in last incremental synchronization will not be update until next full synchronization.

Workaround: You can enable the adapter's retry strategy by configuring the advanced parameters. In the link Advanced tab by default this feature is disabled, and specifying a "retry times" other than 0 will enable this feature. If you don't specify "retry times", then the feature is disabled. If you do not specify the "retry interval", then the default is 10 seconds. This mitigates the locking issue on the SM side. The administrator needs to configure the QCS mail alert so that when there is an error during the synchronization the administrator is notified. The administrator can then reset the timestamp one day earlier or simply do a full sync to resync the failed record. Note that a full sync can be very slow, so try to reset the timestamp whenever possible.

The administrator should schedule a full sync at large intervals (once a month) to maintain data consistency. For more details on "resetting the timestamp" and "full sync" refer to the QCS User's Guide.

Moving a Synchronized Requirement Stop Synchronization

If a requirement is created from an SM change, the requirement is stored in the requirement folder specified by the link. If the requirement is moved from this folder, the requirement is no longer synchronized.

QC 4-byte Integer Limitation

QC only supports 4-byte signed integers (decimal -2,147,483,648 to 2,147,483,647). A non-integer sent to QC from SM will be converted to integer (maximum 4 bytes). Note that

- Any fractional remainder is discarded. For example, if SM sends 12.34 to QC, QC receives 12.
- If more than 4 signed bytes, the upper bits are discarded. For example, if SM sends decimal 4 294 967 299 (hex 1 00 00 00 03), then QC receives 00 00 00 03 (decimal 3).

QC Date Field

The QC date field type contains only date information (for example: 2005-09-09). When a date/time field in SM is mapped to a field in QC, the time information is not synchronized.

QC Special Characters

QC does not allow the following symbols in the requirement name:

- ^
- *
- \

Do not map a SM field containing those characters to QC requirement name (synchronization will fail).

Non-English Characters in a Synchronizer Report

Non-English characters in a synchronizer report are sometimes displayed incorrectly. Refer to Notes and Limitations in <http://updates.merc-int.com/qualitycenter/qc90/sync/qcsynchronizer/QCSyncReadme.htm>.

Service Manager Fields with Non-English Names Not Supported

Do not setup a mapping based on a SM field with non-English name. Synchronization of this kind of mapping may fail.

SM Integration User Account and Password Not Validated

The SM integration user account and password are not validated in a connectivity test and integrity check on QCS. Due to a QCS known issue, user account and password now is only validated in during synchronization start.

B Error Messages

This appendix describes the following categories of error messages:

- *Installation* on page 77
- *Configuration* on page 78
- *Runtime* on page 79
- *XML Validation* on page 81

#MSG_ID

#Message

#Cause

#Solution

Installation

MSG_ID	Message	Cause	Solution
INS_1	Stub for Service Manager Change generation failed.	The Change WSDL URL is not valid.	Provide the valid Change WSDL URL. (i.e. http://localhost:13080/sc62server/PWS/QCIntChangeService.wsdl).
INS_2	The stub for Service Manager Problem generated failed.	The Problem WSDL URL is not valid.	Please provide the valid Problem WSDL URL. (i.e. http://localhost:13080/sc62server/PWS/QCIntChangeService.wsdl).

Configuration

MSG_ID	Message	Cause	Solution
CFG_1	Can not select "Change Management" and "Problem Management" from the endpoint 2 type.	SM adapter cannot be loaded successfully.	Ensure the stub, adapter and dependency jars are in C:\Program Files\HP\Quality Center Synchronizer\adapters\lib.
CFG_2	No create/delete event on change entity is allowed.	For "Change->Defect" and "Change->Requirement" in the Events tab, 'Create a / Delete its corresponding record in the other endpoint' for Endpoint 1 Events is selected.	Select "Do nothing" in the radio-box.
CFG_3	No delete event on the problem entity.	In the Events tab, 'Delete its corresponding record in the other endpoint' is selected.	Select "Do nothing" for all Deletion (Full Synchronization Only) in Events tab.
CFG_4	Missing connection parameter: UserName.	'User name' is empty.	Enter User name in Connectivity tab.
CFG_5	Missing connection parameter: Service URL.	'Service URL' is empty.	Enter Service URL in Connectivity tab.
CFG_6	Missing connection parameter: QC Project.	'QC Project' is empty.	Enter QC Project name in Connectivity tab.
CFG_7	Missing connection parameter: Service URL.	'Service URL' is empty.	Enter Service URL in Connectivity tab.
CFG_8	Connection parameter: Configuration File Path is not valid.	File path is invalid and specified file does not exist.	Enter valid configuration file path name in Connectivity tab (or leave empty).
CFG_9	To connect to endpoint of type SM ChangeManagement. Error: com.hp.qc.synchronizer.adapters.exceptions.AdapterException: Fail to connect to SM:Connection refused: connect. ERROR #2- Fail to connect to SM:Connection refused: connect.	SM Server is shutdown or not available.	Start the SM Server or make available.
CFG_10	ERROR #1- adapter.CONNECTION_FAILURE : Failed to connect to endpoint of type SM ChangeManagement. Error: com.hp.qc.synchronizer.adapters.exceptions.AdapterException: Fail to connect to SM:The web service of SM is not reachable! ERROR #2- Fail to connect to SM:The web service of SM is not reachable!	Web service is not available (for example, is not configured).	Make the web service available.

MSG_ID	Message	Cause	Solution
CFG_11	ERROR #1- adapter.CONNECTION_FAILURE : Failed to connect to endpoint of type SM ChangeManagement. Error: com.hp.qc.synchronizer.adapters.exc eptions.AdapterException: The URL of SM web service is not valid! ERROR #2- The URL of SM web service is not valid!	URL format is wrong.	Correct the URL. The format is: http://<sm server>:<port>/ sc62server/PWS/ [QCIntChangeService QCI ntProblemService].wsdl
CFG_12	Retry times must be an integer between 0 and 3. (0 means disabled).	The value for the paramter "Retries on Locked Recod" in "Advanced" tab is out of scope (0~3).	Input an integer (0 ~ 3) for this paramter "Retries on Locked Recod" in "Advanced" tab.
CFG_13	Retry interval must be an integer between 1 and 10.	The value for the paramter "Retry Interval(Seconds)" in "Advanced" tab is out of scope (1~10).	Input an integer (1 ~ 10) for this paramter "Retry Interval(Seconds)" in "Advanced" tab.

Runtime

MSG_ID	Message	Cause	Solution
RUN_1	Required field <Field Name> can not be empty or SPACE filled.	Synchronized null /space value to a required field from sponsor to receiver.	Ensure that required field values are not null or filled with spaces.
RUN_2	Error when reading web service response from SM: Not authorized	Insufficient rights for SM Integration user when creating/updating defect/ requirement in QC or running synchronization to create/ update the corresponding change/problem.	Provide required rights.
RUN_3	Error 23scxmlapi(23) - XML DOM exception caught - code 5 msg An invalid or illegal XML character is specified	Synchronize with an illegal WSDL caption.	Correct WSDL configuration.
RUN_4	Update failed 1/2/3, retry in 10 seconds, error message=...	Record is locked in SM.	Close the locked record.
RUN_5	Update failed for 3 times, skip, error message=...	Record is locked in SM.	Synchronize record manually or run full synchronization to run all missing updates.
RUN_6	Cannot get field for <Field Name>.	SM adapter cannot get a field.	Ensure stub jar has been generated correctly.

MSG_ID	Message	Cause	Solution
RUN_7	value cannot be reached for <Field Name>.	SM adapter cannot find this field from the stub class.	Ensure stub jar has been generated correctly.
RUN_8	Mapping error, no such property <property name> defined in type <type name>.	SM adapter cannot find this property.	Ensure stub jar has been generated correctly.
RUN_9	Error during setting value for key <key name> with value <value>.	Dynamic model cannot find this key.	Ensure stub jar has been generated correctly.
RUN_10	<Module builder class name> cannot be created because of <message>.	SM adapter can't load a specified class.	Ensure stub jar has been generated correctly.
RUN_11	Can not convert to <target class name> from value <value>.	Value cannot be converted to target type.	Ensure WSDL does not expose unsupported data types.
RUN_12	Exception when getting SM response, return code: <return code>.	Problem with SM communication.	Refer to references for error messages.
RUN_13	Error when reading web service response from SM, record is locked [changeID=<recordID>], message=<Message>.	Record is locked.	Close the locked record.
RUN_14	The data in the '<field name>' field of record <record id> - of file <file name> contains data that does not conform to the SOA data type in datadict.	SOAP field data type in WSDL is not correct. If field is Number type, the value in the database is out of the scope of the specified SOAP type. For example, when choosing IntType (data range: (-2,147,483,648 to 2,147,483,647) for a Number field, if this field has a a value of 2,147,483,648 (2^31) , it will cause this error when reading the record via web service interface.	If this field is Number type, choose "DecimalType" in WSDL. Otherwise select correct SOAP type.

XML Validation

MSG_ID	Error Message	Cause	Solution
XML_1	Failed to validate the configuration file: cvc-elt.1: Cannot find the declaration of element 'test'.	Root element is not "mapping".	Add root element "mapping".
XML_2	Failed to validate the configuration file: cvc-complex-type.2.4.b: The content of element 'mapping' is not complete. One of '{module}' is expected. cvc-complex-type.2.4.b: The content of element 'mapping' is not complete. One of '{module}' is expected.	No "module" element in the "mapping" element.	Add "module" element in the "mapping" root element.
XML_3	Failed to validate the configuration file: cvc-complex-type.2.4.d: Invalid content was found starting with element '{module}'. No child element is expected at this point.	More than 2 module elements in the mapping file.	"mapping" element has only have one or two "module" elements.
XML_4	Failed to validate the configuration file: cvc-enumeration-valid: Value 'others' is not facet-valid with respect to enumeration '[change, problem]'. It must be a value from the enumeration. cvc-attribute.3: The value 'others' of attribute 'name' on element 'module' is not valid with respect to its type, 'ModuleName'.	Name of module is not "problem" or "change".	"name" attribute of "module" element should be "change" or "problem".
XML_5	Failed to validate the configuration file: cvc-complex-type.2.4.b: The content of element 'module' is not complete. One of '{field}' is expected.	No "field" element in the "module" element.	Define "field" elements in each "module" element.
XML_6	Failed to validate the configuration file: cvc-enumeration-valid: Value 'Unkown' is not facet-valid with respect to enumeration '[String, Number, Date, Attachment, Single_Value_List, Multi_Value_List]'. It must be a value from the enumeration. cvc-attribute.3: The value 'Unkown' of attribute 'type' on element 'field' is not valid with respect to its type, 'FieldType'.	"field" element with the wrong type attribute.	Type attribute of "field" element must be enumeration '[String, Number, Date, Attachment, Single_Value_List, Multi_Value_List]'
XML_7	Failed to validate the configuration file: cvc-datatype-valid.1.2.1: 'wrong' is not a valid value for 'boolean'. cvc-attribute.3: The value 'wrong' of attribute 'readonly' on element 'field' is not valid with respect to its type, 'boolean'.	"field" element with the wrong readonly attribute.	"readonly" attribute of field elements should be "true" or "false".
XML_8	Failed to validate the configuration file: cvc-enumeration-valid: Value 'wrong' is not facet-valid with respect to enumeration '[mandatory, optional, recommended]'. It must be a value from the enumeration. cvc-attribute.3: The value 'wrong' of attribute 'required' on element 'field' is not valid with respect to its type, 'FieldRequired'.	"field" element with the wrong required attribute.	"required" attribute of "field" elements should be "mandatory", "optional" or "recommended".

MSG_ID	Error Message	Cause	Solution
XML_9	Failed to validate the configuration file: cvc-complex-type.4: Attribute 'type' must appear on element 'field'.	"field" element without the type attribute.	"type" attribute must be defined in "field" element.
XML_10	Failed to validate the configuration file: cvc-complex-type.4: Attribute 'name' must appear on element 'field'.	"field" element without the name attribute.	"name" attribute must be defined in "field" element.
XML_11	Failed to validate the configuration file: cvc-complex-type.2.4.d: Invalid content was found starting with element 'items'. No child element is expected at this point.	"field" element with more than one child element "items".	Only one "items" element can be defined in each "field" element.
XML_12	Failed to validate the configuration file: cvc-complex-type.2.4.b: The content of element 'items' is not complete. One of '{item}' is expected.	"items" element without the child element "item".	Add "item" element in "items" element.
XML_13	Failed to validate the configuration file: cvc-complex-type.4: Attribute 'value' must appear on element 'item'.	"item" element without the attribute "value".	Define "value" attribute for each "item" element.
XML_14	Failed to validate the configuration file: cvc-minLength-valid: Value " with length = '0' is not facet-valid with respect to minLength '1' for type 'Item'. cvc-complex-type.2.2: Element 'item' must have no element [children], and the value must be valid.	"item" element without text value.	Define text value for each "item" element.
XML_15	cvc-minLength-valid: Value " with length = '0' is not facet-valid with respect to minLength '1' for type 'NonEmptyString'. cvc-attribute.3: The value " of attribute 'value' on element 'item' is not valid with respect to its type, 'NonEmptyString'.	"value" attribute with empty value.	Define value of "value" attribute in "item" element.
XML_16	cvc-minLength-valid: Value " with length = '0' is not facet-valid with respect to minLength '1' for type 'NonEmptyString'. cvc-attribute.3: The value " of attribute 'name' on element 'field' is not valid with respect to its type, 'NonEmptyString'.	"name" attribute with empty value.	Define value of "name" attribute for "item" element.
XML_17	cvc-enumeration-valid: Value 'Attachment' is not facet-valid with respect to enumeration '[String, Number, Date, Single_Value_List, Multi_Value_List]'. It must be a value from the enumeration. cvc-attribute.3: The value 'Attachment' of attribute 'type' on element 'field' is not valid with respect to its type, 'FieldType'.	"field" element with type "Attachment".	Remove Attachment type element.
XML_18	Fail to validate the configuration file: cvc-datatype-valid.1.2.1: 'xxx' is not a valid value for 'integer'. cvc-attribute.3: The value 'xxx' of attribute 'length' on element 'field' is not valid with respect to its type, 'positiveInteger'.	"field" element with incorrect length attribute value.	Correct value of "field" element.