HP Defects and Requirements Exchange with HP Service Manager and HP Application Lifecycle Management

Software Version: 1.03 For the supported Windows[®] operating system

Installation and Administration Guide

Document Release Date: September 2015 Software Release Date: September 2015



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 Notice

© 2008-2015 Hewlett-Packard Development Company, L.P.

Trademark Notices

Adobe® is a trademark of Adobe Systems Incorporated. AMD and the AMD logo are trademarks of Advanced Micro Devices, Inc. Intel and Pentium are trademarks or registered trademarks of Intel Corporation in the United States, other countries, or both. Java and Oracle® are registered trademarks of Oracle and/or its affiliates. Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation. UNIX® is a registered trademark of The Open Group.

Documentation Updates

The title page of this document 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: https://softwaresupport.hp.com

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to: http://h20229.www2.hp.com/passport-registration.html

Or click the New users - please register link on the HP Passport login page.

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

Visit the HP Software Support site at: https://softwaresupport.hp.com.

This website provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support website 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 to sign in. Many also require a support contract. To register for an HP Passport ID, click **Register** on the HP Support site or click **Create an Account** on the HP Passport login page.

To find more information about access levels, go to: https://softwaresupport.hp.com/web/softwaresupport/access-levels.

HPSW Solutions Catalog accesses the HPSW Integrations and Solutions Catalog portal website. This site enables you to explore HP Product Solutions to meet your business needs, includes a full list of Integrations between HP Products, as well as a listing of ITIL Processes. The URL for this website is https://softwaresupport.hp.com/group/softwaresupport/search-result/-/facetsearch/document/KM01702710.

Contents

Chapter 1: Introduction	9
Document Purpose	
Synchronization Concepts	10
References	11
Chapter 2: Planning the Deployment	
Supported Products and Platforms	12
Deployment Scenarios	
Data Types	13
Deployment Tasks	13
Release Package	13
Chapter 3: Installing and Configuring QC/ALM Synchronizer	15
Installing QC/ALM Synchronizer	15
Installation Requirements	16
Deploying the Adapters	
Generating/Deploying the Stub	17
Copying SM Adapter Configuration Files	17
Editing the Configuration Files	18
Module Types	
Field Types	
Requirements	
SM Field Type and Definition Rule SM Change Management Example	
SM Problem Management Example	
Chapter 4: Configuring Links in QC/ALM Synchronizer	24
Create a Link	
Advanced Parameters	28
QC/ALM Field <-> SM Field Mappings	29

Creating Mappings	29
General Mapping Requirements	
Matching Types	31
List Value Mappings	
Constant -> SM Field Mappings	35
Chapter 5: Integration Account	
Creating an SM Integration Account	
Create a Contact Record	
Create a Profile Record	
Create a SecRole Record	
Create an Operator Record	40
Creating a QC/ALM Integration Account	42
Chapter 6: SM Change -> QC/ALM Defect	49
Customizing Service Manager for Change Management	
Add Fields	
Specify the External Access Definition on Service Manager	
Create a Subform	
Add the Subform to a Form	
Add Format Control Calculations/Validations	
Add Rule Set Calculations/Validations	58
Copy an Existing Workflow	
Associate an Existing Change Category with the New Workflow	
Create New Rule Set for Initialization and Validation	60
Associate the New Workflow with the New Rule Set	62
Customizing the QC/ALM Defects Module	64
On QC 10 or Earlier	64
Add Fields	65
Add Tabs	65
Add Fields to Tabs	67
Verify	68
On ALM 11	70
Add fields	
Verify	71

Configuring Links in QC/ALM Synchronizer	73
Specify Endpoints / Type of Link	73
Define Field Mappings	74
QC/ALM Field <-> SM Field	75
Define Events	
Test the Link	
Chapter 7: SM Change -> QC/ALM Requirement	81
Customizing Service Manager for Change Management	81
Customizing the QC/ALM Requirements Module	81
On QC 10 or Earlier	81
Add Fields	82
Add Tabs	83
Add Fields to Tabs	85
Create the SM Incoming Changes Folder	
On ALM 11	
Add Fields	86
Create the SM Incoming Changes Folder	
Configuring Links in QC/ALM Synchronizer	
Specify Endpoints / Type of Link	
Define Field Mappings	
QC/ALM Field <-> SM Field	
Define Events	
Test the Link	92
Chapter 8: SM Problem -> QC/ALM Defect	95
Customizing Service Manager for Problem Management	95
Add Fields	
Specify the External Access Definition on Service Manager	96
Create a Subform	101
Add the Subform to a Form	103
Add Format Control Calculations/Validations	105
Add Rule Set Calculations/Validations	
Copy an Existing Workflow	
Associate an Existing Problem Category with the New Workflow	
Create New Rule Set for Initialization and Validation	

Associate the New Workflow with the New Rule Set	110
Customizing the QC/ALM Defects Module	112
On QC 10 and Earlier	112
Add Fields	112
Add Tabs	113
Add Fields to Tabs	114
On ALM 11	116
Add Fields	116
Configuring Links in QC/ALM Synchronizer	118
Specify Endpoints / Type of Link	118
Define Field Mappings	118
Define Events	119
Test the Link	
Chapter 9: QC/ALM Defect -> SM Problem	122
Customizing Service Manager for Problem Management	
Add Fields	
Specify the External Access Definition on Service Manager	
Create a Subform	127
Add the Subform to a Form	
Customizing the QC/ALM Defects Module	130
On QC 10 and Earlier	130
Add Fields	131
Add Tabs	131
Add Fields to Tabs	
Create a View	
Verify	135
On ALM 11	135
Add Fields	136
Create a View	
Verify	138
Configuring Links in QC/ALM Synchronizer	139
Specify Endpoints / Type of Link	139
Define Filters	
Define Field Mappings	140

QC/ALM Field <-> SM Field	143
Constants -> SM Fields	144
Define Events	146
Test the Link	146
Chapter 10: SM Problem <-> QC/ALM Defect	
Customizing Service Manager for Problem Management	
Add Fields	
Specify the External Access Definition on Service Manager	
Create a Subform	
Add the Subform to a Form	
Add Format Control Calculations/Validations	
Add Rule Set Calculations/Validations	
Copy an Existing Workflow	162
Associate an Existing Problem Category with the New Workflow	
Create New Rule Set for Initialization and Validation	
Associate the New Workflow with the New Rule Set	
Customizing the QC/ALM Defects Module	
On QC 10 and Earlier	
Add Fields	
Add Tabs	
Add Fields to Tabs	
Create a View	173
Verify	173
On ALM 11	174
Add Fields	
Create a View	176
Verify	176
Configuring Links in QC/ALM Synchronizer	
Specify Endpoints / Type of Link	
Define Filters	
Define Field Mappings	178
Define Events	
Test the Link	
Chapter 11: Upgrading the Integration	

Upgrading to the Latest Release	186
Back up Jar Files and Links	186
Deploy the Latest Adapters	186
Upgrade for Various Synchronization Scenarios	187
Upgrade in Non-PD Environment	187
SM Change -> QC/ALM Defect	188
SM Change -> QC/ALM Requirement	189
SM Problem -> QC/ALM Defect	190
QC/ALM Defect -> SM Problem	190
SM Problem <-> QC/ALM Defect	193
Upgrade from Non-PD Environment to PD Content Pack 9.30.2 (for SM 9.3x only) \ldots	193
Upgrade from Non-PD Environment to PD Content Pack 9.30.3/SM 9.4x Codeless	193
Upgrade from 9.3x with PD Content Pack 9.30.2 to 9.30.3/SM 9.4x Codeless/SM 9.4x Hybrid	195
Hybrid	196
Hybrid	196 198
Hybrid Important Notice When Upgrading to SM 9.41 and Above Post-Upgrade Tasks	196 198 199
Hybrid Important Notice When Upgrading to SM 9.41 and Above Post-Upgrade Tasks Appendix A: Error Messages	196 198 199 200
Hybrid Important Notice When Upgrading to SM 9.41 and Above Post-Upgrade Tasks Appendix A: Error Messages Required Fields	196 198 199 200 201
Hybrid Important Notice When Upgrading to SM 9.41 and Above Post-Upgrade Tasks Appendix A: Error Messages Required Fields Installation	196 198 199 200 201 201
Hybrid Important Notice When Upgrading to SM 9.41 and Above Post-Upgrade Tasks Appendix A: Error Messages Required Fields Installation Configuration	196 198 200 201 201 201 204

Chapter 1: Introduction

This chapter describes:

- "Document Purpose" below
- "Synchronization Concepts" on the next page
- "References" on page 11

Document Purpose

This document describes how to configure and deploy the integration components

- HP Service Manager / HP ServiceCenter (SM)
- HP Quality Center (QC)
- HP Quality Center Synchronizer (QCS)

Note: As of version 11.00, HP Quality Center is known as HP Application Lifecycle Management (ALM); as of version 1.40, HP Quality Center Synchronizer (QCS) is known as HP ALM Synchronizer.

This document also describes how to configure and test synchronization links between QC/ALM and SM.

Note: This document contains numerous examples that use the SM and QC/ALM 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 Administrators who must set up and maintain QC/ALM Synchronizer, ensuring that the synchronizer meets all user organization procedural requirements. This document assumes that the reader is an experienced user of either Service Manager or QC/ALM (but not necessarily of both), and therefore describes only the basics of both SM and QC/ALM.

Synchronization Concepts

This section provides a detailed introduction to the following basic synchronization concepts.

• SM Change -> QC/ALM Defect

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

During the QA process, key information is synchronized from QC/ALM 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) in order for the business owner to view updated (scheduled) information in SM. The information includes the status of all changes in the testing cycle.

SM Change -> QC/ALM Requirement

The requirement synchronization feature of this integration allows requirements 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/ALM. 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.

SM Problem -> QC/ALM Defect

After a problem is created, if the CPE engineer determines that there is bug with the problem after analyzing it, and the bug fixing work needs to be tracked, the CPE engineer triggers/initiates the creation of the QC CR ticket. When this problem is marked as "Synchronize with QC Defect", a defect is created in QC/ALM.

QC/ALM Defect -> SM Problem

The business process for defect management in QC/ALM supports creation of known errors in SM based on information in QC/ALM. However, in the current solution, the integration can only create a

problem in SM from a defect in QC/ALM. A user must create the known error in SM manually from the problem in SM. In non-PD environment, known errors are a source of information for informal knowledge articles in the Knowledge Base. In PD environment, known error is a problem record in a special category "known error".

SM Problem <-> QC/ALM Defect

This user story is a combination of SM Problem -> QC/ALM Defect and QC/ALM Defect -> SM Problem.

References

- HP Quality Center Synchronizer User's Guide / HP ALM Synchronizer User's Guide
- HP Quality Center Administrator's Guide / HP ALM Administrator's Guide
- HP Service Manager Installation Guide
- HP Service Manager Online Help
- Best Practices for Publishing and Consuming Web Services with ServiceCenter

Chapter 2: Planning the Deployment

This chapter describes the following deployment plannings:

- "Supported Products and Platforms" below
- "Deployment Scenarios" below
- "Data Types" on the next page
- "Deployment Tasks" on the next page
- "Release Package" on the next page

Supported Products and Platforms

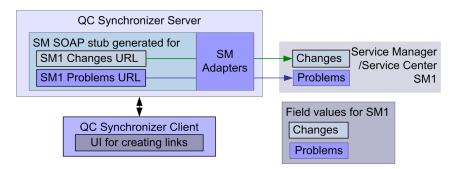
For the supported versions of HP Service Manager, HP Application Lifecycle Management, and HP ALM Synchronizer, refer to the **Support Matrix** section on Defects and Requirements Exchange (SM <-> ALM).

Deployment Scenarios

The deployment scenarios include:

- A single SM server has a dedicated synchronizer.
- A single SM server can connect to multiple QC/ALM systems.

The scenarios are shown in the following figure:



Data Types

"Matching Types" on page 31 describes the data type requirements between QC, QCS, and SM.

Deployment Tasks

Deployment tasks include:

- Customizing Service Manager/Service Center
- Customizing Quality Center/ALM
- Installing/Configuring QC/ALM Synchronizer
- Configuring Links in QC/ALM Synchronizer

Release Package

The release package is delivered as an executable self-extracting installer. Run the installer by double clicking it. The major 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\out-of-box	Out-of-box demo package		

Caution: The out-of-box demo package does not support Service Manager 9.20 or later, ALM 11, or

Installation and Administration Guide Chapter 2: Planning the Deployment

ALM Synchronizer 1.4.

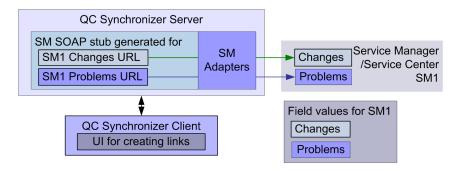
Chapter 3: Installing and Configuring QC/ALM Synchronizer

The QC Synchronizer (QCS)/ALM Synchronizer 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:

- 1. "Installing QC/ALM Synchronizer" below
- 2. "Deploying the Adapters" on the next page
- 3. "Generating/Deploying the Stub" on page 17
- 4. "Copying SM Adapter Configuration Files" on page 17
- 5. "Editing the Configuration Files" on page 18

The following diagram provides an overview of the installation process.



Installing QC/ALM Synchronizer

Download QC Synchronizer from:

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

Download ALM Synchronizer 1.4 from:

http://update.external.hp.com/qualitycenter/qc110/sync/almsynchronizer/index.html

Caution: When installing the ALM Synchronizer, select the ALM 11 mode so that it supports synchronization with ALM 11.

Installation Requirements

The following requirements apply for all supported versions of Service Manager / ServiceCenter, QC / ALM, and QCS / ALM Synchronizer:

- The QC/ALM client is installed when you are logging into QC/ALM. The correct QC/ALM client should be installed on the QC/ALM Synchronizer server.
- The QC/ALM Synchronizer machine should have the same time zone with the QC/ALM machine. For more information, see the integration solution release notes.
 For installation instructions, see the QC/ALM Synchronizer user guide. You can get the user guide from the installation package or from http://h20230.www2.hp.com/selfsolve/manuals.
- Make sure that the time difference in UTC between SM and QC/ALM Synchronizer is within 5 minutes, otherwise the data might be lost during synchronization. For example, the SM server time is 2008-1-1 21:00:00 in UTC, then the QCS server time must be between 2008-1-1 20:55:00 and 2008-1-1 21:05:00.

Deploying the Adapters

Copy all files under the <release-package>\adapter directory to the <QCS_Install_Dir>\adapters\lib directory. Adapters include:

sm-adapter-XX.XX.XXX.jar (XX.XX.XXX is the version number for the current release)

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

sm-adapter-commons-codec-1.3.jar

sm-adapter-commons-httpclient-3.1.jar

Generating/Deploying the Stub

To generate and deploy the stub:

- 1. Start the SM service (stub generation requires access to SM).
- 2. Edit the 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

#Comment this line by this sign "#" if you do not generate stub jar for change management module sm.change.wsdl=http://localhost:13080/sc62server/PWS/ QCIntChangeService.wsdl

#Comment this line by this sign "#" if you do not generate stub jar for problem management module sm.problem.wsdl=http://localhost:13080/sc62server/PWS/ QCIntProblemService.wsdl

- 3. Run the script build.bat from the operating system's command prompt (check the console output for errors). The stub <Release_Package>\build\sm-adapter-ws-client.jar is generated.
- 4. Copy the stub to the <QCS_Install_Dir>\adapters\lib directory.

Copying SM Adapter Configuration Files

To copy SM adapter configuration files:

1. Start/restart QCS.

Click Start > All Programs > HP Quality Center Synchronizer > Start/Stop Synchronizer. The directories <QCS_Install_Dir>\adapters\dat\SM ChangeManagement and SM ProblemManagement appear after the synchronizer service is started (this can take up to one minute).

🗁 C:\Program Files\HP\Quality Center Synchronizer\adapters\dat				
Address 🛅 C:\Program Files\HP\Quality Center Synchronizer\adapters\dat				
ClearQuest C QC RequisitePro SM ChangeManagement SM ProblemManagement				

- 2. Copy the file <Release_Package>\sample\configuration_file_default.xml to the following folders:
 - SM ChangeManagement
 - SM ProblemManagement

Editing the Configuration Files

Edit the files as described below. The files will be specified later when you create links.

Module Types

There are two module types for this configuration file: change or problem.

A module named change or problem means that this module is for Change Management or Problem Management, respectively.

For example:

```
<itg:module name="change">
```

You can define one module or two in this file, but duplicate definitions are not permitted.

Field Types

Field XML Element Specification

Parameter	Description
name	Field name. This name should be the same as Caption enabled in the SM/SC WSDL. This field is required.

Parameter	Description
type	Field type. Its value can be String/Number/Date/Single_Value_List/Multi_Value_List. This field is required.
readonly	Indicates whether the field is read-only. Its value can be true or false. This field is optional. Default is false.
required	Indicates whether the field is mandatory, recommended or optional. This field is optional. Default value for the field is optional.
length	The length of the field in the SM endpoint. This field is optional. The length is unlimited if not specified.

Field XML Element Specification, continued

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 the SM WSDL configuration form, such as Status, Priority)
- Field types
 - String
 - Number
 - Date
 - Single_Value_List
 - Multi_Value_List
- List types
 - Array (multi-value list)
 - 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).

Requirements

- Default field configuration is readable and writable with unlimited length.
- Default field configuration for a Single_Value_List or a Multi_Value_List must be explicitly specified.

- Read-only fields must be explicitly specified.
- You must specify the type and read/write explicitly only for a Single_Value_List/Multi_Value_List.
- If a field is not configured, the field is read/write with unlimited length.
- A list or multi-list field may 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, it must be mapped to Multi_Value_List.
- If the QC field is User_List, 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, the field is mandatory for creation of a new entity.
- If the field has a length limitation (attribute length), values from other endpoints could be truncated to match this limitation.

SM Field Type and Definition Rule

There are restrictions on data type and field type combination. Define field type in the configuration file according to such rules.

SM 7.0x/7.10 DB data type	SC 6.2 DB data type	Field type on Form	WSDL data type	Field Definition Rule
Date/time	Date/time	Date	DateTimeType	Optional. Permitted type is "Date".
Number	Decimal	Decimal Text	< <i>Empty</i> > DecimalType IntType	Optional. Permitted type is "Number".
Logical	Boolean	Check Box Radio Button	< <i>Empty</i> > BooleanType	Optional. Permitted type is "String".
Character	Text	Text TextArea Combo Box Comfill	< <i>Empty</i> > StringType	Optional. Permitted types include "String" and "Single_Value_List".

SM	Field	Туре	and	Definition	Rules
----	-------	------	-----	------------	-------

SM 7.0x/7.10 DB data type	SC 6.2 DB data type	Field type on Form	WSDL data type	Field Definition Rule
Array	Array	Text Area	<empty></empty>	Optional. Permitted types include "String" and "Single_Value_List".
Array	Array	Combo Box Comfill Text	<empty></empty>	Required. Permitted type is "Multi_ Value_List".

SM Field Type and Definition Rules, continued

SM Change Management Example

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" type="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>

</itg:mapping>

SM Problem Management Example

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">
```

Installation and Administration Guide Chapter 3: Installing and Configuring QC/ALM Synchronizer

<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">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: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>

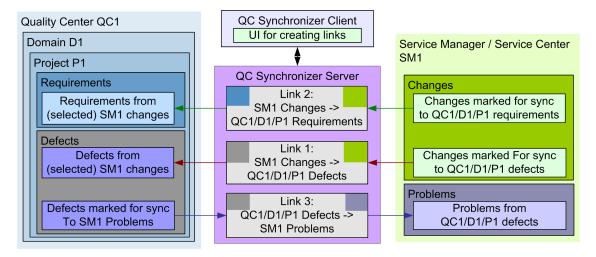
</itg:module>

</itg:mapping>

Chapter 4: Configuring Links in QC/ALM Synchronizer

This chapter describes how to configure and test links. Although instructions in this chapter are provided as examples using QC Synchronizer and Quality Center, they still apply for ALM Synchronizer and ALM.

The following diagram summarizes link configuration:



You need to create synchronization links in QC/ALM Synchronizer between two endpoints. Each endpoint is an application or system containing data that is synchronized by the synchronizer. A link defines which entities are included in the synchronization, and how the synchronization is performed.

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

- "Create a Link" on the next page
- "QC/ALM Field <-> SM Field Mappings" on page 29
- "List Value Mappings" on page 35
- "Constant -> SM Field Mappings" on page 35

Filters are only required for QC/ALM Defect -> SM Problem (see "Define Filters" on page 140). The events settings determine what the synchronizer 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.

Note: A link cannot be duplicated. For example, if a link already exists for SMServer1/Changes -> QCServer1/Domain1/Project1/Defects, a second link between these two entities cannot be created.

End Point	Parameter	Requirements
QC	Username	
QC	Password	
QC	Server URL	
QC	Domain	
QC	Project	
SM	User name	
SM	Password	
SM	Service URL	http:// <service_manager_host>: < port >/sc62server/PWS/QCIntChangeService.wsdl or http://<service_manager_host>: < port >/sc62server/PWS/QCIntProblemService.wsdl</service_manager_host></service_manager_host>
SM	Adapter Configuration (SM field values) filename	Empty or the adapter data folder file (see "Copying SM Adapter Configuration Files" on page 17).
SM	QCProject	Required (because of an adapter limitation). The format is <qc_host>/<qc_domain>/<qc_project></qc_project></qc_domain></qc_host>

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).

🥳 Create Link - Step 1 of 4 - General Properties						
	Assign gener	al properties:				
	Link name:	change_to_defect_1				
(QP)	Description:	change to defect 1				
Quality Center Synchronizer	Endpoint 1 type:	Quality Center				
Synchronizer	Endpoint 2 type:	SM ChangeManagement 💌				

- 3. Click Next. The "Step 2: Assign QC endpoint connection properties" dialog appears.
- 4. Enter the required information.

飛 Create Link - Step 2	of 4 - Quality Center Endpoi	nt 🗵
STITULA		
Ø		iter endpoint connection properties:
Quality Center	User name: SMQCIntUse	21
Synchronizer	Password:	
	Parameter	Value
	ServerURL	http://localhost:8080/gcbin
	Domain	DEFAULT
	Project	Demo
		Chec <u>k</u> Connectivity

- 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).

🧝 Create Link - Step 2	of 4 - Quality Center Endpoint	X	
3 Trans			
Quality Center Synchronizer	Assign SM ChangeMa	nagement endpoint connection prop	
	Parameter	Value	
	Service URL	http://localhost:13080/sc62server/PW5/QCIntChangeService.w	isdl
	Configuration File Name	configuration_file_default.xml	
	QC Project	localhost/DEFAULT/Demo	
		Chec <u>k</u> Connectivity	

Note: QC Project has the same value as specified on SM customization.

- 7. Click **Next**. If this is a change management link, "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 on the **Connectivity** tab.

:			SM ChangeMar	nagement		
Jser name:	SMQCIntUser		User name:	SMQCIntUser		
Password:			Password:	•••••		
Parameter		Value	Parameter		Value	
Domain		DEFAULT	Configuration	File Name	configuration_file_default.xml	
Project		Demo	QC Project		localhost/DEFAULT/Demo	
ServerURL		http://localhost:8080/gcbin	Service URL	September 1	http://localhost:13080/sc62serv	

Advanced Parameters

Advanced parameters are shown on the Advanced tab.

Parameter	Value
Socket Timeout (Minutes)	2
Retries On Locked Record	0
Retry Interval (Seconds)	10

Retries On Locked Record

When a record in the SM endpoint is locked, it will cause synchronization failure. The integration will retry the synchronization according to the value of this parameter. 0 means the retry feature is disabled. For details, see *HP Defects and Requirements Exchange with HP Service Manager and HP Quality Center Release Notes*.

• Retry Interval (Seconds)

When the retry feature is enabled, this parameter defines the retry interval. The retry interval must be an integer between 1 and 10. For details, see *HP Defects and Requirements Exchange with HP Service Manager and HP Quality Center Release Notes*.

• Socket Timeout (Minutes)

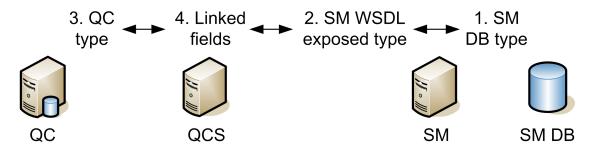
Socket connection will be established during synchronization. If there are many records matching the filter in the SM endpoint, retrieving list operation will cost some time, which might cause timeout of socket connection. This parameter is used to define the socket timeout. Its range is: 0-120.

QC/ALM Field <-> SM Field Mappings

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

- "Creating Mappings" below
- "General Mapping Requirements" on the next page
- "Matching Types" on page 31

The following diagram shows the field mapping chain.



For examples of field mappings, see:

- SM Change to QC/ALM Defect, " QC/ALM Field <-> SM Field" on page 75.
- SM Change to QC/ALM Requirement, "QC/ALM Field <-> SM Field" on page 91.
- QC/ALM Defect to SM Problem, "QC/ALM Field <-> SM Field" on page 143.

Creating Mappings

To map fields:

- 1. Select a field on each side.
- 2. Select a direction in the Map Selected Fields tab.

P Map Selected Fields	-								
Create bidirection	nal mapping					🙉 🗇 🔿 Add Co			
Map QC field to S		_	SM Chang	eManagement Chan	ge as Defect !	Schema			
+ Map SM Changel	anagement field to	OC field	apped		Name		Туре	Attributes	Mapped
a r nony	origio raiso				Chang	eNumber	String	RW/	No
Problem ID	String	RW	No		Pa Descri	ption	String	RW/	No
Project	Single value	list RW	No		Nodifie	ed	Date	RW/	No
Reproducible	Single value	list RW	No		CEnt	iyiD	Number	RW/	No
Severity	Single value	list RW	No		CPro	ject	String	RW/	No
Status	Single value	list RW	No	1	Urgena	cy.	Single value	list RW	No
Summary	String	RW/	No	•					
× 🎸 Check Field Map	oping 🕘 Import (∋ Export				Mapping Properties	Value Mapp	ing Field Prop	verties
Aapped Fields						Misc			
Type QC Field Direction		SM ChangeMa Field	nageme	int	Direction Dominant side		<> Endpoint 1		
Summary		>	Description			Synchronize bac	k on create 1	No	

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 cannot 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 other field.

Matching Types

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

Data Type Combinations

QC Data type	QCS QC Type	Dir	QCS SM type	WSDL data type	Field type on form in SM/SC	SM DB data type for SM	SM DB data type for SC
Number	Number	<->	Number	DecimalType or IntType ¹	Decimal or Text	Number	Decimal
String	String	<->	String	BooleanType	Check Box or Radio Button	Logical	Boolean
Date ²	Date	<->	Date	DateTimeType (required)	Date	Date/time	Date/time
String	String	<->	String ³	StringType	Text, TextArea, Combo Box or Comfill	Character	Text
Memo	Memo/ String	<->	String ⁴	StringType	Text, TextArea, Combo Box or Comfill	Character	Text
User List ⁵	User List	->	String ⁶	StringType	Text, TextArea, Combo Box or Comfill	Character	Text

Data Type Combinations, continued

QC Data type	QCS QC Type	Dir	QCS SM type	WSDL data type	Field type on form in SM/SC	SM DB data type for SM	SM DB data type for SC
Lookup List	Single- value list	<->	Single- value list/ String ⁷	StringType	Text, TextArea, Combo Box or Comfill	Character	Text
String	String	<->	String	StringType	TextArea	Array ⁸	Array ⁹
Memo	Memo String	<->	String	StringType	TextArea	Array ¹⁰	Array ¹¹
Lookup List	Single- value List	<->	String	StringType	TextArea	Array ¹²	Array ¹³
User List ¹⁴	User List	<->	String	StringType	TextArea	Array ¹⁵	Array ¹⁶
Lookup List	Multi- value List	<->	Multi- value List	StringType	Text, Comfill or Combo Box	Array ¹⁷	Array ¹⁸
Attachment ¹⁹	Attachment	<->	Attachment	Attachment	Attachment	lmage	lmage

Note: There is no need to explicitly specify WSDL data type on WSDL configuration for all types except for the Date type. For details, see *Best Practices for Publishing and Consuming Web Services with ServiceCenter*.

¹IntType supports a data range from -2,147,483,648 to 2,147,483,647.

²QC data only supports Yr/Mo/Dt.

³It 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.

⁴See footnote 3.

Installation and Administration Guide Chapter 4: Configuring Links in QC/ALM Synchronizer

⁵Write 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.

⁶See footnote 3.

⁷See footnote 3.

⁸Only an array of characters is supported.

⁹See footnote 8.

¹⁰See footnote 8.

¹¹See footnote 8.

¹²See footnote 8.

¹³See footnote 8.

¹⁴See footnote 5.

¹⁵See footnote 8.

¹⁶See footnote 8.

¹⁷See footnote 8.

¹⁸See footnote 8.

¹⁹ The SM-QC/ALM integration does not support synchronization of an attachment with OKB size. Consider the following guidelines when mapping attachment fields:

- You can create only one mapping between attachment fields per link.
- Synchronizer identifies attachments by their file name, and not by their content. Therefore:
 - If you change the file name of an attachment, even if you do not change its content, Synchronizer determines that the original attachment has been deleted and a new attachment added, and synchronizes the attachment fields accordingly.
 - If you have different attachments in each of the endpoints, but they have the same file name, Synchronizer is not able to distinguish between them and considers them as the same attachment.
- For a bidirectional attachment field mapping, if an attachment was updated in both endpoints since the last synchronization, Synchronizer copies the attachment in the non-dominant endpoint to the conflict_backup directory, located under the main HP ALM Synchronizer directory. It then overwrites the attachment in the non-dominant endpoint with the attachment in the dominant endpoint.

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-values lists, and therefore the values must be specified in an XML file.

Some list fields also require mapping of available values (as shown in the following figure).

Map Selected Fields					N ChangeManager			ema	
Name	Туре	Attributes	Mapped	-	Name	Tj	pe	Attributes	Mapped
Priority	Single value list	RW/	No		ChangeNumber	St	ing	RW/	No
Problem ID	String	RW/	No		Description	St	ing	RW	No
Project	Single value list	RW	No		Modified	D-	xe.	RW	No
Reproducible	Single value list	RW/	No		CEntityID	No	mber	RW/	No
Seventy	Single value list	RW	No		CProject	St	ing	RW	No
Status Status	Single value list	RW/	No		Urgency	Si	ngle value list	RW/	No
Summary	String	RW/	No	-					
apped Fields				м	apping properties	Value ma	pping Fie	ld propert	ies
Sevenity	<>	Urgeno	ay .	E	ndpoint 1 value	Direction	Endpoint	2 value	
				5	-Urgent	<>	1 - Critica		
				4	Very High	<>	2 · High		
				3	High	<>	3 - Avera	ge	

Constant -> SM Field Mappings

For examples of constant -> SM field mappings, see "Constants -> SM Fields" on page 144.

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

ieneral	Connectivity Scheduling	Filters Events	Field Mapping
Mappe	d Fields		
Туре	Endpoint 1 Field	Direction	Endpoint 2 Field
8	Value: AUTO	>	AssignmentGroup
8	Value: client system	>	Category
3	Value: BOB.HELPDESK	>	ProblemOwner

Chapter 5: Integration Account

This chapter includes:

- "Creating an SM Integration Account" below
- "Creating a QC/ALM Integration Account" on page 42

Creating an SM Integration Account

The integration account is equivalent to an operator in Service Manager for exclusive use with this solution.

This section includes:

- "Create a Contact Record" below
- "Create a Profile Record" below
- "Create an Operator Record" on page 40

Create a Contact Record

Create a contact for the integration administrator by clicking **System Administration**> **Base System Configuration** > **Contacts** in Service Manager; or clicking **Support** > **Contacts** in ServiceCenter.

Page	Field	Value
Contact Information	Contact Name	<administrator's name=""></administrator's>
Contact Information	Full Name	<administrator's full="" name=""></administrator's>

Create a Profile Record

To create a profile record for the integration account:

• Create profile for Change Management to synchronize SM Changes with QC/ALM Requirements and Defects ("Change -> Requirement" and "Change -> Defect").

Profile records grant specific rights and privileges to the integration account to enable Change Management.

On Service Manager:

Click **System Administration > Ongoing Maintenance > Profiles**and create a Change management profile record by 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	Yes	Check Box
5	Security/Rights	Reopen	Yes	Check Box
6	Query	Query Options	Yes	Check Box

On ServiceCenter:

Click **Services** > **Change Management** > **Maintenance** > **Profiles** and create a change management profile record with the parameters shown in the following table.

No	Tab Page	Field	Value	Comment
1		Profile Name	CMProfile_QCInt	
2		Profile Area	Changes	
3	Basic/Basic Options	Open	Yes	Check Box
4	Basic/Basic Options	Reopen	Yes	Check Box
5	Basic/Basic Options	Save	Yes	Check Box
6	Query/Query Options	All	Yes	Check Box

Create profile for Problem Management to synchronize SM Problem with ALM/QC Defect ("Problem <-> Defect", "Problem -> Defect" and "Problem <- Defect").

Profile records grant specific rights and privileges to the integration account to enable Problem

Management.

On Service Manager:

Click **System Administration > Ongoing Maintenance > Profiles**and create a change management profile record with the parameters shown in the following table.

No	Tab Page	Field	Value	Memo	Remarks
1		Profile Name	PMProfile_ QCInt		
2	Problems/Security/Rights	New	Yes	Check Box	This parameter is not required when synchronizing SM Problems with QC/ALM Defects ("Problem - > Defect").
3	Problems/Security/Rights	Close	Yes	Check Box	
4	Problems/Security/Rights	Update	Always		
5	Problems/Security/Rights	Reopen	Yes	Check Box	

On ServiceCenter:

Click **Services** > **Problem Management** > **Administration** > **User Profiles**and create a problem management profile record with the parameters shown in the following table.

No	Tab Page	Field	Value	Memo	Remarks
1		Profile Name	PMProfile_ QCInt		
2	Problem Details	Browse	Yes	Check Box	
3	Problem Details	Open	Yes	Check Box	This parameter is not required when synchronizing SM Problems with QC/ALM Defects ("Problem -> Defect").

No	Tab Page	Field	Value	Memo	Remarks
4	Problem Details	Update	Yes	Check Box	
5	Problem Details	Reopen	Yes	Check Box	

Create a SecRole Record

Note: Steps in this topic are applicable for the Service Manager Process Designer (PD) Content Pack 9.30.x only. Skip this topic if the PD Content Pack is not installed.

You can create a SecRole Record for Change Management to synchronize SM Changes with QC/ALM Requirements and Defects ("Change -> Requirement" and "Change -> Defect"). The SecRole records grant specific rights and privileges to the integration account to enable Change Management.

To create a SecRole record for the integration account:

- 1. Click System Administration > Security > Roles.
- 2. Create a role named SMQCIntSecRole with the parameters shown in the following table. Remove all rights from all areas with the exception of the Change area and the Problem area.

No	Tab Page	Field	Value	Comment
1		Security Role Name	SMQCIntSecRole	
2		Security Area	Change	
3	Rights	View	Yes	Check Box
4		Update	Always	
		Security Area	Problem	
5	Rights	View	Yes	Check Box
6		New	Yes	Check Box
7		Update	Always	

Create an Operator Record

The operator record identifies the logon name, password, and other settings for each SM operator. Create the required operator records by clicking **System Administration** > **Ongoing Maintenance** > **Operators** on Service Manager; or clicking **Utilities** > **Administration** > **Security** > **User Administration** > **Search for Operators** on ServiceCenter with the parameters shown in the following table.

No	Page	Field	Value	Remarks
1	General	Logon Name	SMQCIntUser	
2	General	Full Name	QC Integration Default Account	
3	General	Contact ID	<integration administrator's account in SM></integration 	The contact created in the previous section.
4	Security	Unlimited Sessions	Yes	Check Box
5	Security	Password	<your password=""></your>	
6	Startup	Execute Capabilities	SOAP API	
7	Login Profile	Time Zone	Greenwich/Universal (or create a time zone with no time difference or DST switch in Database Manager)	
8	Login Profile	Date Format	yy/mm/dd	The date format cannot be changed (changing it will cause loss of all data during synchronization).
9	Startup	Execute Capabilities	ChMAdmin	Set the two parameters to synchronize SM Changes with QC/ALM Requirements and
10	General/Application Profiles	Change Profiles	CMProfile_QCInt	Defects ("Change -> Requirement" and "Change -> Defect").

Operator Record Parameters

No	Page	Field	Value	Remarks
11	Startup	Execute Capabilities	ProbAdmin	Set the two parameters to synchronize SM Problem with ALM/QC Defect ("Problem <->
12	General/Application Profiles	Problem Profile	PMProfile_QCInt	Defect", "Problem -> Defect" and "Problem <- Defect").

Operator Record Parameters, continued

Note: If the Service Manager Process Designer (PD) Content Pack 9.30.x is installed, refer to the parameters shown in the following table for the General/Application Profiles configuration.

Operator Record Parameters

No	Page	Field	Value	Remarks
10	General/Application Profiles	Security Role	SMQCIntSecRole	Set the this parameter to synchronize SM Changes with ALM/QC Requirements and Defects ("Change -> Defect" and "Change -> Requirement").
12	General/Application Profiles	Security Role	SMQCIntSecRole	Set this parameter to synchronize SM Problems with ALM/QC Defects ("Problem <-> Defect", "Problem -> Defect"and "Defect -> Problem").

Creating a QC/ALM Integration Account

To create an integration account:

1. Create a User.

Perform the following steps:

- a. Log in to the "Quality Center Site Administration" or "Application Lifecycle Management Site Administration" using the QC/ALM site administrator account.
- b. On the Site Users tab, create and configure integration account SMQCIntUser (including the User Name and password).
- c. In the Site Projects tab, choose the project from the list.
- d. Click the Project Users tab in the right panel, and click **Add From The Users List**.
- e. Add the configured user SMQCIntUser to the project.
- f. Log off.

2. Create a Group.

Perform the following steps:

On Quality Center 10 and Earlier:

- a. Log on to the QC project using a project administrator account.
- b. Click **TOOLS** > **Customize...**.
- c. Select Groups.
- d. Click New.
- e. Enter name SMIntegration.
- f. For Create As: select Viewer.
- g. Click **OK**.
- h. Select **Yes** to create the user group.

On ALM 11:

- a. Log on to the ALM project using a project administrator account.
- b. Click **TOOLS** > **Customize...**.
- c. Select Groups and Permissions.
- d. Click New Group.
- e. Click **Yes** to create the new group.
- f. Enter name **SMIntegration**.

- g. For Set As: select Viewer.
- h. Click **OK**.

3. Assign Permissions.

Perform the following steps:

a. Click the SMIntegration group, go to the Permissions tab, and assign permissions for the user group on the Requirements, Defects, and Administration subtabs as shown in the following tables.

• Change -> Requirement

Subtab	Permission	QC Setting	ALM Setting
Subtab	Permission Add Requirement Modify Requirement	QC Setting Image: Add Requirement Image: Delete Requirement Image: Delete Requirement Image: Add Tests To Coverage Image: Remove Tests From Coverage Image: Add Requirement Traceability Image: Modify Requirement Traceability Image: Remove Rem	ALM SettingSelect the following options:• Coverage Level • Create• Update• Requirement • Create• Update• Risk-Based Quality Management • Assess Business Criticality• Assess Failure Probability
			Assess Functional Complexity

• Change -> Defect

Subtab	Permission	QC Setting	ALM Setting
--------	------------	------------	-------------

Defects	Add Defects/ Modify Defects	✓ Add Defect ● ✓ Modify Defect □ Delete Defect □ Add Defect Link ● □ Modify Defect Link ● □ Remove Defect Link	 Select the following options: Defect Create Update
---------	--------------------------------	--	---

• Problem -> Defect

Subtab	Permission	QC Setting	ALM Setting
Defects	Add Defects/ Modify Defects	Add Defect Modify Defect Delete Defect Add Defect Link Modify Defect Link Remove Defect Link	Select the following options: • Defect • Create • Update

• Problem <- Defect

Subtab	Permission	QC Setting	ALM Setting
Defects	Add Defects/ Modify Defects	Add Defect Modify Defect Delete Defect Add Defect Link Modify Defect Link Remove Defect Link	Select the following options: • Defect • Create • Update

Add Public	Add Public Favorite Views	Select the following options:
Favorite Views	Modify Public Favorite Views	
	Delete Public Favorite Views	Add Public Favorite View Folders
•	Add Private Favorite Views	Add Public Favorite Views
Favorite views	Modify Private Favorite Views	
Delete Public	🔽 Delete Private Favorite Views	Delete Public Favorite View Folders
Favorite Views		• Delete Public Favorite Views
Add Private		Manage Private Favorite Views
Favorite views		
Modify Private		 Manage Project Planning and Tracking
Favorite Views		• Modify Public Favorite View Folders
Delete Private		Modify Public Foyorite Views
Favorite Views		Modify Public Favorite Views
	Favorite Views Modify Public Favorite Views Delete Public Favorite Views Add Private Favorite Views Modify Private Favorite Views Delete Private	Favorite ViewsModify Public Favorite ViewsModify Public Favorite ViewsImage: Constraint of the second secon

• Problem <- Defect

		QC setting	ALM setting
Defects	Add Defects/ Modify Defects	Add Defect Modify Defect Delete Defect Add Defect Link Modify Defect Link Remove Defect Link	Select the following options: Defect Create Update

Add Public	Add Public Favorite Views	Select the following options:
Favorite View	Modify Public Favorite Views	
Madifu Dublia	Delete Public Favorite Views	Add Public Favorite View Folders
2	Add Private Favorite Views	Add Public Favorite Views
	Modify Private Favorite Views	
Delete Public	🔽 Delete Private Favorite Views	• Delete Public Favorite View Folders
Favorite Views		Delete Public Favorite Views
Add Private Favorite Views		Manage Private Favorite Views
Modify Private		Manage Project Planning and Tracking
Favorite Views		Modify Public Favorite View Folders
Delete Private Favorite Views		Modify Public Favorite Views
	Favorite View Modify Public Favorite Views Delete Public Favorite Views Add Private Favorite Views Modify Private Favorite Views Delete Private	Favorite ViewModify Public Favorite ViewsModify Public Favorite ViewsDelete Public Favorite ViewsDelete Public Favorite ViewsAdd Private Favorite ViewsAdd Private Favorite ViewsAdd Private Favorite ViewsModify Private Favorite ViewsDelete Private Favorite ViewsDelete Private Private Favorite ViewsDelete Private Private Private Private Private PrivateDelete Private Private Private Private Private Private Private Private Private Private Private Private Private Private Private Private Private Private

- b. Add the integration user SMQCIntUser to group SMIntegration.
- c. Save and close. The integration account is created.

Chapter 6: SM Change -> QC/ALM Defect

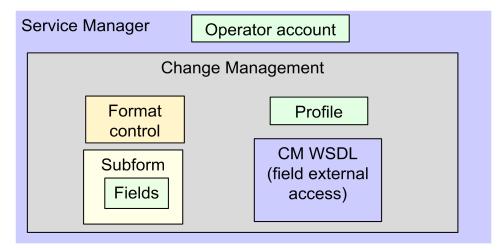
This chapter describes how to synchronize SM Changes with QC/ALM Defects ("Change -> Defect").

This chapter includes:

- "Customizing Service Manager for Change Management" below
- "Customizing the QC/ALM Defects Module" on page 64
- "Configuring Links in QC/ALM Synchronizer" on page 73

Customizing Service Manager for Change Management

The following diagram summarizes the components which require tailoring in Service Manager.



To customize Service Manager for Change Management, perform the following tasks:

- 1. "Add Fields" on the next page
- 2. "Specify the External Access Definition on Service Manager" on the next page
- 3. "Create a Subform" on page 52

- 4. "Add the Subform to a Form" on page 54
- 5. "Add Format Control Calculations/Validations" on page 56

Add Fields

To add the required fields:

- 1. Click System Definition > Tables > cm3r.
- 2. Add the following required fields to the cm3r table. Do not change them.

	Туре			
Field	Service Manager	ServiceCenter		
qcintegration.type	Character	Text		
qcintegration.id	Number	Decimal		
qcintegration.project	Character	Text		

Note: The data type requirements for SM fields are described in "Matching Types" on page 31.

Specify the External Access Definition on Service Manager

To specify the External Access Definition on Service Manager:

- Create a custom External Access Definition QCIntChangeService by clicking Tailoring > WSDL configuration on Service Manager 7.0x; or clicking Tailoring > Web Services > Web Service Configuration on Service Manager 7.1x or later with the following values:
 - Service Name: QCIntChangeService
 - Name: cm3r
 - Object Name: QCIntChange
 - Allowed Actions: save / Action Names: Update

object.name Change QCIntChang								
External Ad	cess I	Def	inition					
Service Name	QCIntC	:han	igeService					
Name:	cm3r			•	Obje	ct Name:	QCIntChange	
🗇 Allowed Actions 🛛 🧇			Expressions	4	Fields			
Allowed Actions			Action Nam	es		Action T	уре	
save		Update						
3070								

Note: The above values are required (Do NOT change them).

2. Enable required fields in the web service.

Field	Caption	Туре
header,number	ChangeNumber	StringType
qcintegration.id	QCEntityID	IntType
sysmodtime	Modified	DateTimeType
affected.itemsm.device.display.name	ServiceName	
middle,logical.namesm.device.display.name	ConfigurationItemName	
middle,assetssm.device.display.name	AssetsName	

object.name							
Change							
QCIntChange							
External Ac	cess	Definitio	n				
Service Name:	QCInt	ChangeSer	vice				
Name:	cm3r			•	Object	Name:	QCIntChange
Allowed A	ctions	🔶 Expre	ssions	4	> Fields		
Field			Captio	n		Tγ	′pe
header, num	nber		Change Number		St	ringType	
qcintegratio	in.id		QCEntityID		In	tType	
severity			Urgency		St	ringType	
sysmodtime			Modified		Da	ateTimeType	
header,rea	son		Reason		Í		
description.	structu	re,desc	Description				
middle,logic	middle,logical.name			ConfigurationItem			
header,risk.assessment			RiskAssessment				
header,coordinator			Coord	inal	tor		
header,requested.by			Reque	ste	dBy		
header,prio	rity.co	de	Priorit	Y .			

Note: The affected.item..sm.device.display.name field, the middle,logical.name..sm.device.display.name field, and the middle,assets..sm.device.display.name field are available for Service manager 9.41 only.

Note: The caption value must be unique and alphanumerics (no spaces) with the first letter capitalized (AValidCaption123, AnotherValidCaption and so on). The above values are required (Do NOT change them).

Create a Subform

To create a subform:

1. Create a global list by clicking **Tailoring > Tailoring Tools > Global Lists** on Service Manager; or clicking **Utilities > Tools > Global Lists** on ServiceCenter) with the following parameters:

No.	Parameter	Value	Remarks
1	List Name	SMQC Integration CM Project List	
2	Regen Every	1 00:00:00	
3	Build List on Startup?	Yes	Check box
4	List Variable	\$G.qcintegration.change.project	
5	User Defined List?	Yes	Check box
6	Value List	{"server1/domain1/project1", "server2/domain2/project2"}	Change to the values for your system.
			Note: No spaces between slashes.

Save this global list and click **Rebuild Global List** in the Options menu.

2. Click **Tailoring** > **Forms Designer** on Service Manager; or click **Toolkit** > **Forms Designer**on ServiceCenter to create the cm3r.qcint.subform subform with the following components:

Note: Click No when the system message "Do you want to use Form Wizard?" appears.

Component	Properties
Label	Caption: Forward to QC:
Combo Box	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:

Component	Properties			
Text	Input: qcintegration.id			
	Read-Only: Yes			
Label	Caption: Server/Domain/Project:			
Combo Box	Input: qcintegration.project			
	Value List: \$G.qcintegration.change.project			
	Read-Only Condition: [\$qcint.project.readonly]			
	Mandatory Condition: [qcintegration.type]>0			

😽 Forms Designer: cm3r.qcint.subform 🗙					
Forward to QC:		×			
Def/Req ID:					
Server/Domain/Project:					
🔲 Properties 🗙	📄 Properties 🗙				
Property	Value	Current Entries			
Input Read-Only Condition	qcintegration.type [\$qcint.readonly]	0 - Not Forward 1 - Forward as Requirement 2 - Forward as Defect			
Display List					

Add the Subform to a Form

If the Service Manager Process Designer (PD) Content Pack 9.30.2 or 9.30.3 is not installed, follow the steps below to add the subform you created to a form:

Note: The following steps also apply to SM 9.4x Classic.

Note: In Service Manager 9.20 or later, a form may contain only collapsible sections (groups) instead of notebook tabs. If this is the case, replace the terms "notebook tab" and "tab" with "section" in the following steps.

1. Open the form of a phase of a category by using the Forms Designer (cm3r.rfc.build.g is used as an example).

- 2. Add a notebook tab with the QC Integration caption.
- 3. Add a subform to the new tab with format cm3r.qcint.subform, which is displayed in the following screenshot:

😽 Forms Designer: cm 🗙		Properties 🛛	
Reques	t For Change	Si	ubformat
fieldset1175878835347		Property	Value
RFC No.	Planned Start:	Name	subview1208421811523
Phase	Planned End:	X	14
Status	Risk Assessment:	Y	2
		Width	90
Approval Status	Initial Impact Assessment:	Height	16
Alert Stage:	Urgency:	Visible	\checkmark
Reason for Change:	Priority:	Visible Condition	
		Format	cm3r.qcint.subform
	Folder:	Virtual Join	
	ociated CIs 🗇 CI Info 🗇 Affected	Display Blank	\checkmark
♦ qc integration		Display Using Table	
•		Input	
	_		
	-		
	•		

4. Save the changes.

Note: If the error message "Format 'cm3r.qcint.subform' not found (display, show.rio)" appears, log out and then log in again to enable the subform.

If the Service Manager Process Designer (PD) Content Pack 9.30.2 or 9.30.3 is installed, follow the steps below to add a subform:

Note: The following steps also apply to SM 9.4x Codeless/Hybrid.

- 1. Open the form of a phase of a category by using the Forms Designer (chm.normal.registration is used as an example).
- 2. Add a notebook tab with the QC Integration caption.
- 3. Add a subform to the new tab with format cm3r.qcint.subform, which is displayed in the following screenshot:

To Do Queue: My To Do List 🛛 📓 Forms Design	er 🛃 Forms Designer: chm.normal.registration 🗵	Properties 🛛	
🖥 OK 🗯 Cancel		85 -	Subformat
4 🖹 🖬 🗂 🗂 🗂 🗐 🖉 🌐	9 -	📅 🖾 🛅 螽 🔝 🐂 📔 Property	Value
		Name	
		X	16
		Y	12
		Width	64
		Height	22
		Visible	V
Workflow 🗇 Affected Services 🗇 Associated C	Is 🗇 Tasks 🗇 🗇 SLA 🗇 Attachments 🧇 QC Integration 🗇 Closure 🗇 QC Integration 🎽	% Visible Condition	
		Tab Stop	0
		Format	cm3r.qcint.subform
		Virtual Join	
		Display Blank	×
		Display Using Table	
		Input	

3. Save the changes.

Note: If the error message "Format 'cm3r.qcint.subform' not found (display, show.rio)" appears, log out and then log in again to enable the subform.

Add Format Control Calculations/Validations

Note: Steps in this topic are applicable if the Service Manager Process Designer (PD) Content Pack is not installed, or if you are working with SM 9.4x Classic. Otherwise, skip this topic and refer to "Add Rule Set Calculations/Validations" on page 58 for configurations in the PD environment and the SM 9.4x Codeless/Hybrid environment.

To add format control calculations and validations, follow the steps below:

- Open the format control record of the previous change form by clicking Tools > Format Control on ServiceCenter 6.2, or clicking Tailoring > Format Control on Service Manager (cm3r.rfc.build is used as an example in Service Manager 7.0x)
- 2. Click Calculations.
- 3. Add two rows with the following values.

display	initial	calculation
true	true	<pre>\$qcint.type.readonly=2;if (qcintegration.type in \$file~=0) then (\$qcint.type.readonly=1)</pre>
true	true	<pre>\$qcint.project.readonly=2;if (qcintegration.type in \$file~=0 and not null (qcintegration.project in \$file)) then (\$qcint.project.readonly=1)</pre>

Note: When you copy the calculations into the rows, make sure that each calculation is in one line; also note that there is a space between lines in the table above. For example, the calculation in the first row is: \$qcint.type.readonly=2;if (qcintegration.type in \$file~=0) then (\$qcint.type.readonly=1)

The Change calculations are shown in the following figure:

Forms		Querie	es	Calc	lations JavaScript Validations Subroutines	Addl Options Privileges
					Format Control Maintenance - Calculations	
Name	:		cm3r.rfe	c.build		View: short
add	update	delete	disp	initial	calculation	
true	true				risk.assessment in \$file=nullsub(risk.assessment in \$file, "1")	
true	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	<pre>\$qcint.readonly=2;if (qcintegration.type in \$file="1" or qcintegration.type in \$file="2") then (\$qcint.readonly=1)</pre>	

4. Click Validations.

5. Add a row with the following values.

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

The Change validation values are shown in the following figure:

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

6. Save the changes.

Add Rule Set Calculations/Validations

Note: Steps in this topic are applicable for Service Manager 9.3x with Process Designer (PD) Content Pack 9.30.2 or 9.30.3 and Service Manager 9.4x Codeless/Hybrid. Otherwise, refer to "Add Format Control Calculations/Validations" for configurations in SM 9.3x non-PD environment and SM 9.4x Classic.

To add rule set calculations and validations, follow the steps below:

- 1. "Copy an Existing Workflow" below
- 2. "Associate an Existing Change Category with the New Workflow" on the next page
- 3. "Create New Rule Set for Initialization and Validation" on page 60
- 4. "Associate the New Workflow with the New Rule Set" on page 62

Copy an Existing Workflow

You can use copies of the existing workflows in another business process, or make changes to the HP proprietary workflow copies.

To copy an existing workflow, follow the steps below:

- 1. From the System Navigator, click Tailoring > Process Designer > Copy Existing Workflow.
- 2. On the Clone a Workflow page, select the workflow you want to copy. For example, Normal.
- 3. Type SMQCIntChM in the New workflow name field.

To Do (Queue: My To Do List	Wizard: Clone a Workflow 🗷	
CI	one a Workflov	N	
		Please specify the new workflow name, as well as the p	refix for new rule sets if they are to be copied as well.
		New workflow name: * SMC	1CIntChM
		Copy rule sets?	

- 4. Select the Copy rule sets check box if you want to copy rule sets, and then type a rule set prefix.
- 5. Click **OK**.

The newly copied workflow appears in the list on the Clone a Workflow page.

Associate an Existing Change Category with the New Workflow

You can update existing change categories, subcategories and areas and associate the updated categories with the new workflow so that they can be used in another business process. For more information, refer to *HP Service Manager – Process Designer Content Pack Administrator's Guide*.

To associate an existing change category with the new workflow, follow the steps below:

- 1. From the System Navigator, click **Change Management > Configuration > Change Categories**.
- 2. Click Search.
- 3. Select the change category for which you want to add a workflow. For example, Normal Change.
- 4. In the Change Category Definition page, remove the currently assigned workflow from the Workflow field.
- 5. Type SMQCIntChM in the workflow field.

6. Click **Save** to associate the change category with the workflow.

Change Category Definition		
	Normal Change	
Description	Normal Change	
Availability		
	SMOChitchm	
Default Template		
	Assign number before commit?	
Workflow SubCategories		4
Registration and Categorization > Validation	n \Rightarrow Risk and Impact Analysis 🔿 TCAB Approval 🖚 Build and Test 🔷 DCAB Approval 💠 Deployment 💛 Backout	
Abandoned	CMDB Update -> Post Implementation Review	
	Closure	

Create New Rule Set for Initialization and Validation

To create a new rule set for initialization, follow the steps below:

- 1. From the System Navigator, click **Tailoring > Process Designer > Rule Sets**.
- 2. Type the values as follow:

Field	Value
ID	chm.alm.int.init
Available as action	
Name	Initialize for ALM integration in the Change Record
Table name	cm3r
HP Proprietary	

- 3. Click New and Save.
- 4. Click Add Rule.
- 5. In the Select Rule Type page, click Run JavaScript.

Installation and Administration Guide Chapter 6: SM Change -> QC/ALM Defect

6. In the Run JavaScript page, type the values as follow:

Field	Value
Rule Description	Run Javascript for initializing Integration type and project in the Change Record
Statement	<pre>vars['\$qcint.type.readonly'] = 2; vars['\$qcint.project.readonly'] = 2; var _null=system.functionsnull; var file = vars.\$L_file; if(file["qcintegration.type"] !=0 && !_null(file["qcintegration.type"])) { vars['\$qcint.type.readonly'] = 1 } if(file["qcintegration.type"] !=0 && !_null(file["qcintegration.project"])) { vars['\$qcint.project.readonly'] = 1 }</pre>

7. Click **OK**.

8. Click **Save** and **Exit**.

To create a new rule set for validation, follow the steps below:

- 1. From the System Navigator, click **Tailoring > Process Designer > Rule Sets**.
- 2. Type the values as follow:

Field	Value
ID	chm.alm.int.validation
Available as action	
Name	Validation for ALM integration in the Change Record

Field	Value
Table name	cm3r
HP Proprietary	

- 3. Click New and Save.
- 4. Click Add Rule.
- 5. In the Select Rule Type page, click **Set Mandatory Fields**.
- 6. Click Edit. The Condition Editor opens.
- 7. Add conditions as illustrated in the following screenshot:

Condition Edit	or				×
Match all of the	following conditions				+ =
CurrentRecord	Qcintegration Type	Not Equals	Value	0	+ = -
CurrentRecord	Qcintegration Type	Not Equals	Blank/NULL	NULL	+ = -
					Done Cancel
					Cancel

- 8. Click **Done** to exit the Condition Editor.
- 9. Set value "Qcintegration Project" to "Field Name".
- 10. Click **OK** to exit the Set Mandatory Fields page.
- 11. Click Save and Exit.

Associate the New Workflow with the New Rule Set

To associate the new workflow with the new rule set, follow the steps below:

- 1. From the System Navigator, click **Change Management > Configuration > Change Workflows**.
- 2. Select SMQCIntChM in the workflows list.
- 3. Select the first phase in the workflow graph.
- 4. Click **Rule Sets** tab > **Initialization** tab.
- 5. Click Add and select the chm.alm.int.init rule set you just created.

HP Service Manager							User: falcon
		To Do Queue: My To Do List Workflows	Work	flow: SMQCIntChM 🛞			
🖻 📑 🚭 🔹 «	<	💾 Save 🔍 Zoom in 🍳 Zoom out 🔲 Add	phase	Telete Broperties			8
Request Management Service Catalog Service Desk Service Level Management System Administration Talong > Audd > Differential Upgrade	•	Abandoned	.)	Risk and Impact Analysia	Build and Test		Badout
 Document Engine Event Services 				Rule Sets - Initialization	1		
 knowledge Engineering knottactions Process Designer Configuration Configuration Configuration Cony Exating Workflow Export Workflow Rule Sets Workflows SofL Utities SofL Utities Veta Services Codes Database Dictionary Database 	m	Phase - Registration and Categorization Details Forms Rule Sets Actions On entry On ext Initialization C Add @ Detac GP Yow All Q-QP Dow Rule Sets		id CoseChange apply template chm 00 validation chm aam rt. nit chm aam, rt. nit chm aam, rt. nit chm aam, rt. nit chm aam, change model chm clean. closetime chm closer, closetime chm closer, witzard	Itame Cose Change Cose Change Cose Change Cose Change Com OO Intal com OO Intal com OOV addation Retatict for ALM integration in the Change Record Validation for ALM integration in the Change Record Chan Cose Cose Integration Chan Cose Cose Integration Chan Cose Cose Integrated Chan Cose Integrated Cha	*	Cioure
Data Policy Format Control					OK Cancel	J	

- 6. Click **OK**.
- 7. Click Rule Sets tab > On display tab.
- 8. Repeat step 5 and 6.
- 9. Click **Rule Sets** tab > **On update** tab.
- 10. Click **Add** and select the chm.alm.int.validation rule set you just created.

MP Service Manager				User: falcon
	To Do Queue: My To Do List Workflows	Workflow: SMQCIntChM 🗵		
🖬 🕼 🥸 🔍 👋	💾 Save 🍕 Zoom in 🤤 Zoom out 🔲 Add p	ohase 🐨 Delete 🚰 Workflow properties		브 📄 🗌
Request Management				
Service Catalog				
Service Desk				
Service Level Management		<u> </u>	<u></u>	
System Administration	Registration and Categorization Validation	Risk and Impact Analysis	al 📅 Build and Test> DCAB Approval 👘 Deployment	Backout
Tailoring	اله السب و			
> Audit				
Differential Upgrade	Abandoned		CMDB Up	date Post Implementation Review
Document Engine	Abanooneo	· · · · · · · · · · · · · · · · · · ·	CMDB OP	Date Post Implementation Review
Event Services		Add Rule Sets - On enter		
Knowledge Engineering		🛅 ld	Name	
Notifications		CloseChange	Close Change	Closure
Process Designer		apply.template	Apply Template	
Configuration	•	and the second s		
Copy Existing Workflow	Phase - Registration and Categorization	chm.OO.initial	Chm OO Initial	
Export Workflow		chm.OOvalidation	chm.00Validation	
Rule Sets	Details Forms Rule Sets Actions	chm.alm.int.init	Initialize for ALM integration in the Change Record	
Workflows	On enter On exit Initialization On	Chm.alm.int.validation	Validation for ALM integration in the Change Record	
SQL Utilities		Chm apply change model	Chm Apply Change Model	
Tailoring Tools	🖨 Add 💮 Delete 🙃 View 🏫 Up 🕹 Down	Chm clean closetime	Chm Clean Closetime	
Web Services	Rule Sets	Chm.close.related	Chm Close Related	
Codes		Chm.closure.wizard	Chm Closure Wizard	
Database Dictionary Database Manager		Chill Closure. Wizard	Chill Closure Wizard	T
Database Manager Data Policy			OK Cancel	
Format Control			on ouron	
ronnac control				

- 11. Click **OK**.
- 12. Click **Rule Sets** tab > **On enter** tab.
- 13. Repeat step 10 and 11.
- 14. Click Save.

Customizing the QC/ALM Defects Module

The steps for customizing the Defects module vary with different QC versions.

- "On QC 10 or Earlier" below
- "On ALM 11" on page 70

On QC 10 or Earlier

To customize the Defects module on Quality Center 10 or earlier, perform the following tasks:

- 1. "Add Fields" on the next page
- 2. "Add Tabs" on the next page
- 3. "Add Fields to Tabs" on page 67
- 4. "Verify" on page 68

Add Fields

To add the required fields for Defect module customization:

1. Log on to QC as a project administrator.

D . . .

- 2. Click **Tools / Customize**. The "QC Project Customization" module opens.
- 3. Add the following fields for the defect entity in Project Entities (*XX* and *XY* are sequential numbers auto-generated by QC).

Field Name	Field Label	Field Type
BG_USER_ <i>XX</i>	Change ID	String
BG_USER_ <i>XY</i>	Created from	String

The following figure shows an example project entity.

Quality Center - Project Custor	nization		Domain: DEFAULT, Project: Demo, User: admin
User Properties Project Users Groups Module Access Project Entities Requirement Types Risk-Based Quality Management Project Lists Automail Alert Rules Workflow	Project Entities Project Entities Project Entities Cycle Cy	Field Settings	BG_USER_02 Change ID String ¥ 40 Required Searchable

Note: The data type requirements for QC fields are described in "Matching Types" on page 31.

Add Tabs

To add tabs to the Defect form and show fields on these tabs:

1. In "QC - Project Customization", click **Workflow** > **Script Editor**.

2. Select Defects module script.

Quality Center - Project Customization

User Properties Project Users	Workflow
Groups Module Access Project Entities Requirement Types Risk-Based Coulity Man Project Lists Automal	Script Generator - Add Defect Field Customization Enables you to customize the fields displayed for each user group in the Add Defects dialog bo You can also specify field order and whether a field is required. Script Generator - Defect Details Field Customization Enables you to customize the fields displayed for each user group in the Defect Details dialog box. You can also specify field order and whether a field is required.
Alert Rules Workflow	Script Editor Enables you to write VBScript code for all Quality Center modules. You can also use the Script Editor to modify the scripts generated by the above tools.
Script Editor Script Editor Toolbar Bu	tton Editor GetNewBugPageName
Workflow Scripts Ormon script S	le script

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

Bug_CanDelete
 Bug_AfterPost
 SetFieldApp

select case PageNum

🖲 📓 Manual Runner script

E-S Defects module script

case "2"

GetNewBugPageName = "SM Integration (New)"

end select

Note: The parameter 2 specifies tab 2 (the second tab). For a new bug, the tab name is SM Integration (New).

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

select case PageNum

case "2"

GetDetailsPageName = "SM Integration (Details)"

end select

Note: The parameter 2 specifies tab 2 (the second tab). For an existing defect, the tab name is SM Integration (Details).

Add Fields to Tabs

To add fields to tabs:

- 1. In "QC Project Customization", click **Workflow** > **Script Editor**.
- 2. Select **Defects module script**.

User Propertie Project Users		/orkflow
Module Access Project Entities Requirement Types Risk-Based Quality Man Project Lists Project Lists		cript Generator - Add Defect Field Customization nables you to customize the fields displayed for each user group in the Add Defects dialog bo ou can also specify field order and whether a field is required. cript Generator - Defect Details Field Customization nables you to customize the fields displayed for each user group in the Defect Details dialog ox. You can also specify field order and whether a field is required.
Alert Rules Workflow	E	i <u>cript Editor</u> nables you to write VBScript code for all Quality Center modules.
Script Edit	11	ou can also use the Script Editor to modify the scripts generated by the above tools.
Script Editor	11	

- 3. If **WizardFieldCust_Details** and **WizardFieldCust_Add** are not found in the list, do the following to generate these two methods.
 - a. Script Generator Add Defect Field Customization
 - b. Script Generator Defect Details Field Customization



4. 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

The parameter values are:

- Field name (BG_USER_XX, where XX consists of two digits)
- Visible (True)
- Required (False)
- Page number (start from 0)
- View order (start from 0)
- 5. 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

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

Bug_Fields.Field("BG_USER_XX").IsReadOnly=True

Bug_Fields.Field("BG_USER_XY").IsReadOnly=True

7. Save your changes.

Verify

To verify whether the Defects module on Quality Center 10 or earlier is customized successfully:

🤍 New Def	fect	
× Clear	Attach: 🥒 🥜 📸 🚺 🚔 🕵 🕶 💱 💷 🖤	0
* Summary:		
Details	SM Integration (New)	
	Change ID: Created from:	
	Submit Close	

1. Create a new defect. The dialog box has a new tab titled "SM Integration (New)" with two fields.

2. Open an existing defect. The second tab is titled "SM Integration (Details)", and both the "Change ID" field and the "Created from" field are read-only.

📴 Defect Deta	nils	
	> 🞗 🏲 ! 🗉 🕇 🕄	0
Defect:	13 a new defect	
6	Details SM Integration (Details)	
Details	Change ID: C18	Created from: Created from SM/SC
Û,		
Attachments		
a		
Linked Entities		
\$		
History		
Execution Report	OK Cancel	

On ALM 11

On ALM 11, you only need to add new fields directly to the Details tab of the Defect form.

To customize the ALM Defects module, perform the following tasks:

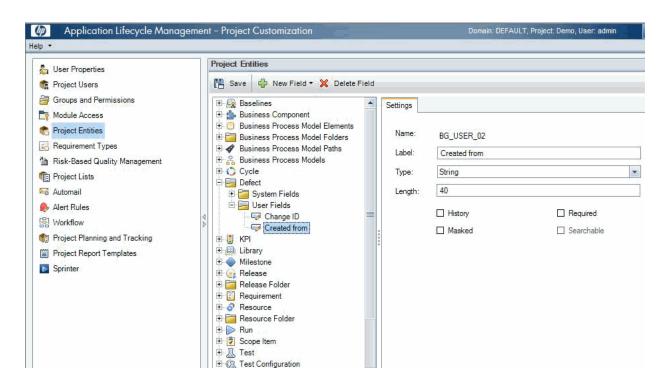
- 1. "Add fields" below
- 2. "Verify" on the next page

Add fields

To add the required fields for Defect module customization:

- 1. Log on to ALM as a project administrator.
- Click Tools / Customize. The "Application Lifecycle Management Project Customization" module opens.
- 3. Add the following fields for the defect entity in project entities (*XX* and *XY* are sequential numbers auto-generated by ALM).

Field Name	Field Label	Field Type
BG_USER_XX	Change ID	String
BG_USER_XY	Created from	String



- 4. Click Workflow > Script Editor.
- 5. Select Defects module script.
- Set the **Readonly** fields by adding the following lines to the **Bug_New** and **Bug_Moveto** subroutines:

Bug_Fields.Field("BG_USER_XX").IsReadOnly=True

Bug_Fields.Field("BG_USER_XY").IsReadOnly=True

7. Save your changes.

Verify

To verify the Defects module on ALM 11 is customized successfully:

New Defect		
🗙 💩 - 🍫 🛃 📇		-
* Summary:		
Details Attachments		
	Submit Close Help	

1. Create a new defect. Both the "Change ID" field and the "Created from" field are read-only.

2. Open an existing defect. Both the "Change ID" field and the "Created from" field are always readonly.

🖳 🗖 🔀						
Defect Details Image: Subscript of the state of the	Details Reproducible: Severity: City City City City City City City City	Medium	Assigned To: Status: Target Cycle: Change ID: Problem ID: ments:	New C10003	dd Comment	
		OK Cancel	Help			

Configuring Links in QC/ALM Synchronizer

To configure and test a link in the QC/ALM synchronizer, perform the following tasks:

- 1. "Specify Endpoints / Type of Link" below
- 2. "Define Field Mappings" on the next page
- 3. "Define Events" on page 78
- 4. "Test the Link" on page 78

Specify Endpoints / Type of Link

Caution: If you are using ALM Synchronizer 1.4 with ALM 11.0, install the HP Quality Center Connectivity Add-in on the ALM server before you proceed; otherwise ALM Synchronizer will not be able to connect to ALM. To install this add-in, click the **Add-Ins Page** link on your ALM options window: http:// <ALM Platform server name> <: port number >/ qcbin.

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

- 1. Step 1: "Endpoint 2 type" = **SM ChangeManagement**.
- 2. Step 2: "Service URL" = http://service_manager_host>:<port>/sc62server/PWS/QCIntChangeService.wsdl
- 3. Step 3: "Select entity types" = Change as Defect.

Define Field Mappings

Basic field mappings are summarized below:

QC	Direction	SM	Constant value	Remarks
Change ID	<-	ChangeNumber		
Defect ID	->	QCEntityID		Synchronize back on create: Yes
Created from			Created from SM/SC	

Example field mappings are shown in the following screenshot:

Mapped Fields					
Туре	QC Field	Direction	SM ChangeManagement Field		
ē	Severity	<>	Urgency		
-	Change ID	<	ChangeNumber		
<mark>-</mark>	Defect ID	>	QCEntityID		
-	Summary	<>	Description		
</th <th>Created from</th> <th><</th> <th>Value: Created from SM/SC</th>	Created from	<	Value: Created from SM/SC		

QC/ALM Field <-> SM Field

The following table summarizes the field mappings between QC/ALM and SM. The first two rows are required mappings.

SM Change - QCIALM Delect Mappings									
QC Len	QC DB Name	QC Type	QC/ QCS Label	QC/ QCS Type	Dir	QCS SM Type	QCS Name/ SM WSDL Caption	SM WSDL Type	SM DB Name/ SM WSDL field
40	BG_USER _02 ²	String	Change ID	String	<-	String	ChangeNumber	StringType	header,number
10	BG_BUG _ID	Number	Defect ID ³	Number	->	Number	QCEntityID ⁴	IntType	qcintegration.id
255	BG_ SUMMARY	String	Summary	String	<->	String	Description	StringType	description.struc description
70	BG_ SEVERITY	Lookup List	Severity	Single value list	<->	Single value list	Urgency	Character	severity

SM Change -> QC/ALM Defect Mappings

If you specify a value mapping, for example, Severity <-> Urgency, you can specify as follows:

QCDe	efect Schem	a	SM ChangeManagement	100		
Name		уре	Name Type			
Pu Se	evenily S	lingle value list	Urgency Single value	i list		
lappe	d Fields			Mapping properties	Value ma	pping Field properties
Туре	QC Field	Direction	SM ChangeManagement	Endpoint 1 value	Direction	Endpoint 2 value
Type	QC FIEID	Direction	Field	5-Urgent	<>	1 - Critical
C1	Change ID	<	ChangeNumber	4-Very High	<>	2 · High
6	Defect ID	>	QCEntityID			
5	Summary	(>	Description	3-High	<>	3 - Average
1.1	No. of Concession, Name	1000		2-Medium	<>	4 - Low
(C)-	Severity	<>	Urgency	C		

The following tables summarize the mappings you just created between the single value lists (which have their own directions).

SM Change	-> Q	C Defect	List	Value	Mappings
-----------	------	----------	------	-------	----------

QC Len	QC DB Name	QC Type/ Lookup list Values	QC/QCS Label	QC/QCS Type	Dir	QCS SM Type	QCS Name/ SM WSDL Caption	SM WSDL Type	SM DB Name/ SM WSDL Field	SM DB Type	SM Len
70	BG_ SEVERITY	Lookup List	Severity	Single value list	<->	Single value list	Urgency	String Type	severity	Char (SM7) or Text (SC6)	40

Detailed value mappings between the QC/QCS Severity field and the QCS/SM WSDL Urgency field are listed in the table below:

QC Value (from Lookup List) ¹	Value Map Dir	SM Field Value (from SM Adapter Config File for Change Management) ²
5-Urgent	<->	1-Critical
4-Very High	<->	2-High

QC Value (from Lookup List) ¹	Value Map Dir	SM Field Value (from SM Adapter Config File for Change Management) ²
3-High	<->	3-Average
2-Medium	<->	4-Low

1

Lookup list is created in QC.

Field Settings		
Field Name:	BG_SEVERITY	Project Lists
Field Label:	Severity	Lists: Severity 💌
Field Type:	Lookup List 💌	List Items
Lookup List		1-Low
		📄 2-Medium
Severity	New List Goto List	📄 3-High
		📄 4-Very High
🗹 Verify Value	3	📄 5-Urgent

²The XML file is in <*QCS_Install_Dir*>\adapters\dat\SM ChangeManagement\configuration_file_default.xml (see "SM Change Management Example" on page 21).

Define Events

The following table lists the event settings for the two endpoints.

Operation	QC Action (Event)	SM Action (Event)
Creation	Do nothing.	Create a corresponding record in the other endpoint.
Update	Update its corresponding record in the other endpoint.	Update its corresponding record in the other endpoint.
Deletion	Do nothing.	Do nothing.

The following screenshot shows the settings:

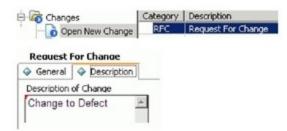
C Links → ⊅ change_to_defect_1	
General Connectivity Scheduling Filters Events Field Mapping Adv	vanced
QC	SM ChangeManagement
Creation	Creation
When a record is created in this endpoint	When a record is created in this endpoint
C Create a corresponding record in the other endpoint	Create a corresponding record in the other endpoint
C Do nothing	Do nothing
Update	Update
When a record is updated in this endpoint	When a record is updated in this endpoint
Update its corresponding record in the other endpoint	C Update its corresponding record in the other endpoint
Do nothing	Do nothing
Deletion (Full Synchronization Only)	Deletion (Full Synchronization Only)
When a record is deleted from this endpoint	When a record is deleted from this endpoint
© Do nothing	© Do nothing
© Delete its corresponding record in the other endpoint	© Delete its corresponding record in the other endpoint
© Recreate based on its corresponding record in the other endpoint	© Recreate based on its corresponding record in the other endpoint

Test the Link

To test the link:

Note: The following sample steps are for your reference only. The exact steps required on your system may differ significantly. The phase in which the QC Integration tab 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 customization; RFC is used in ServiceCenter 6.2/Service Manager 7.0x as an example).



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 a value in the Server/Domain/Project field and select **Forward as Defect** in the Forward to QC field.

Backout Method	History	Approvals	Attachments	Related Records	Workflow	QC Integration	27
	Forward to	QC:		2 - Forward as Defe	t		
	Def/Req ID	:					

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.

📴 Defect Deta	is 📃 🖬	×
	N 🗶 🕨 I 🗉 V 🗊 🤇	
Defect:	9 Change to defect	ן
Details Details Attachments Einked Entities History	Details SM Integration (Details) Change ID: C19 Created from: Created from SM/SC	
Execution Report	OK Cancel	

Note: In ALM 11, the Change ID and Created from fields reside on the Details tab of the Defect form.

Chapter 7: SM Change -> QC/ALM Requirement

This chapter describes how to synchronize SM Changes with QC/ALM Requirements ("Change -> Requirement").

This chapter includes:

- "Customizing Service Manager for Change Management" below
- "Customizing the QC/ALM Requirements Module" below
- "Configuring Links in QC/ALM Synchronizer" on page 88

Customizing Service Manager for Change Management

For details, see the "Customizing Service Manager for Change Management" section in "SM Change -> QC/ALM Defect".

Customizing the QC/ALM Requirements Module

The steps for customizing the Requirements module vary with different Quality Center/ALM versions.

- "On QC 10 or Earlier" below
- "On ALM 11" on page 86

On QC 10 or Earlier

To customize the Requirements module on Quality Center 10 or earlier, perform the following tasks:

- 1. "Add Fields" on the next page
- 2. "Add Tabs" on page 83

- 3. "Add Fields to Tabs" on page 85
- 4. "Create the SM Incoming Changes Folder" on page 87

Add Fields

To add required fields for requirement customization, follow these steps.

- 1. Log on to QC as a project administrator.
- 2. Click **Tools / Customize**. The "QC Project Customization" module displays.
- 3. Add the following fields for the requirement entity in project entities (*XX* and *XY* are sequential numbers auto-generated by QC).

Field Name	Field Label	Field Type
RQ_USER_ <i>XX</i>	Change ID	String
RQ_USER_ <i>XY</i>	Created from	String

Note: The data type requirements for QC fields are described in Matching Types.

This is shown in the following figure.

Project Entities			Project Entities
Project Entities	Field Settings Field Name: Field Label: Field Type: Field Length: History Masked	RQ_USER_01 Change ID in SM String 40 Searchable	Project Entities

 In Requirement Types add fields "Change ID"/"Created from" to the Business type requirement. Business type is the default requirement type for incoming requirements (other types can be used).

Quality Center - Proj	ect Customizatio	n	
User Properties Project Users Groups Module Access Project Entities Requirement Types Risk-Based Quality Mana Project Lists Automail Alert Rules	Requirem Types Folder Functional Group Testing Undefined	ent Types Properties Assigned Icon: Test Coverage: Risk-Based Quality Management: User Fields	Replace None Perform Assessment
Workflow		Not In Type Old Type (obsolete)	In Type > Name Required >> Change ID in SM X

Add Tabs

To add tabs to the Requirement form and display the fields on these tabs, click **Workflow** > **Script Editor**. Add the following code to the requirement module.

Note: For a new Requirement, the tab label is "SM Integration (New)". For an existing Requirement, the tab label is "SM Integration (Details)". The parameter 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.

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"

Installation and Administration Guide Chapter 7: SM Change -> QC/ALM Requirement

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"

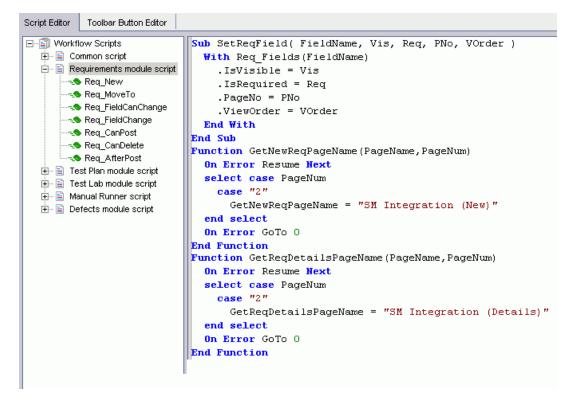
GetReqDetailsPageName = "SM Integration (Details)"

end select

On Error GoTo 0

End Function

The resulting script is shown in the following screenshot:



Add Fields to Tabs

To set the fields as read-only and place the fields on the tabs, 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).

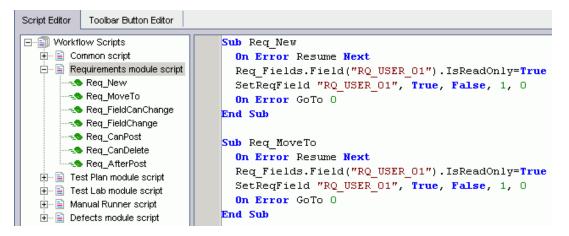
Req_Fields.Field("RQ_USER_XX").IsReadOnly=True

Req_Fields.Field("RQ_USER_XY").IsReadOnly=True

SetReqField "RQ_USER_XX", True, False, 1, 0

SetReqField "RQ_USER_XY", True, False, 1, 1

The resulting script is shown in the following screenshot:



Create the SM Incoming Changes Folder

To create the folder for Requirements originated from SM changes:

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

Requirements	Edit View	Favorites	Analysis		
$\equiv \mathbb{G}_n \times$	J • ₹ &	🛛 📑 🖉	⊕ <u>-</u> ≋ - <u>C</u>	¥.	
🕕 🕄 🚦 🛌 Name	9		•	Direct Cover Status	Reg ID
	Requirement	s			0

On ALM 11

In ALM 11, you only need to add new fields directly to the Details tab of the Requirement form.

To customize the ALM Requirements module, perform the following tasks:

- 1. "Add Fields" below
- 2. "Create the SM Incoming Changes Folder" on the next page

Add Fields

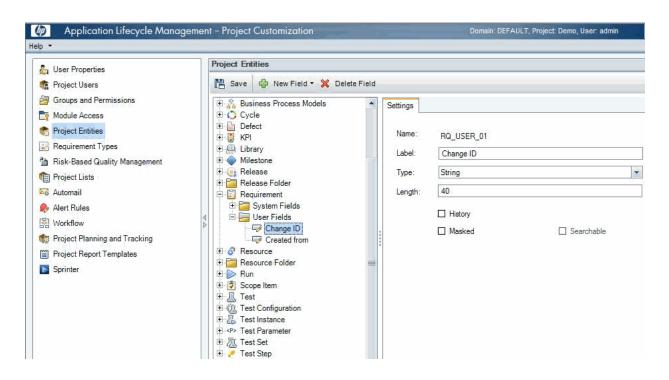
To add required fields for requirement customization, follow these steps.

- 1. Log on to ALM as a project administrator.
- Click Tools / Customize. The "Application Lifecycle Management Project Customization" module opens.
- 3. Add the following fields for the requirement entity in project entities (*XX* and *XY* are sequential numbers auto-generated by ALM).

Field Name	Field Label	Field Type
RQ_USER_ <i>XX</i>	Change ID	String
RQ_USER_ <i>XY</i>	Created from	String

Note: The data type requirements for QC fields are described in "Matching Types" on page 31.

This is shown in the following figure.



- In Requirement Types add fields "Change ID"/"Created from" to the Business type requirement. Business type is the default requirement type for incoming requirements (other types can be used).
- 5. Click Workflow > Script Editor.
- 6. Select Requirements module script.
- 7. 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_XX").IsReadOnly=True

Req_Fields.Field("RQ_USER_XY").IsReadOnly=True

8. Save your changes.

Create the SM Incoming Changes Folder

To create the folder for Requirements originated from SM changes:

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

Configuring Links in QC/ALM Synchronizer

To configure and test a link in QC/ALM synchronizer, perform the following tasks:

- "Specify Endpoints / Type of Link" below
- "Define Field Mappings" on page 90
- "Define Events" on page 92
- "Test the Link" on page 92

Specify Endpoints / Type of Link

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

- 1. Step 1: "Endpoint 2 type" = SM ChangeManagement.
- 2. Step 2: "Service URL" =
 http://service_manager_host>:<port>/sc62server/PWS/QCIntChangeService.wsdl
- 3. Step 3: "Select entity types" = Change as Requirement.

4. Specify the incoming requirement folder as shown in the following screenshot:

General	Connectivity	Scheduling	Filters	Sut	otype Mapping	Advanced
		QCIntUser				
P	arameter		Valu	е		
Do	omain		DEF	AULT		
Pr	oject		Dem	5		
Se	erverURL		http:/	//loca	alhost:8080/qcb	pin
					Check Conne	ectivity
	Use alternate equirements\		:d_incor	ning_	requirement_fol	der>

Requirements will be created in the specified folder in QC.

🕕 🖁 🚦 🛌 Name	
🖃 🚞 Requirements	
🖂 😑 SM Incomin	g Changes
🔳 💼 change	to problem

5. In the "Sub types mapping" tab, specify the type of requirements created from changes. General Connectivity Scheduling Filters Sub Types Mapping

Endpoint 1 A	wailable Sub Types	Mapped Sub Types			
Functional Testing Undefined	Add Mapping	Business · SM Change			
		Req ID: 6 Aname: Change to problem	* Requirement Type:	Business Business Folder	Ż
				Functional	•

Define Field Mappings

Basic field mappings are summarized below:

QC	Direction	SM	Constant value	Remarks
Change ID	<-	ChangeNumber		
Req ID	->	QCEntityID		Synchronize back on create: Yes
Created from			Created from SM/SC	

Example field mappings are shown in the following screenshot:

Mappe	Mapped Fields						
Туре	QC Field	Direction	SM ChangeManagement Field				
- E	Name	<>	Description				
Ð	Change ID	<	ChangeNumber				
- E	Req ID	>	QCEntityID				
$\langle \rangle$	Created from	<	Value: Created from SM/SC				

QC/ALM Field <-> SM Field

The following table summarizes the field mappings between QC/ALM and SM. The first two rows are required mappings.

QC Len	QC DB Name	QC Type	QC/QCS Label	QC/ QCS Type	Dir	QC S SM Type	QCS Name/ SM WSDL Caption	SM WSDL Type	SM DB Name	SM7 DB type	SC6 DB type	SM Len
40	RQ_USER_01	String	Change ID	String	<-	String	ChangeNumber	StringType	header,number	Char	Text	100
10	RQ_REQ_ID	Number	ReqID	Number	->	Number	QCEntityID	IntType	qcintegration.id	Num	Decimal	xx
255	RQ_REQ_ COMMENT	Memo	Description	String	<->	String	Description		description.structure, description	Char	Text	xx

SM Change -> QC/ALM Requirement Mappings

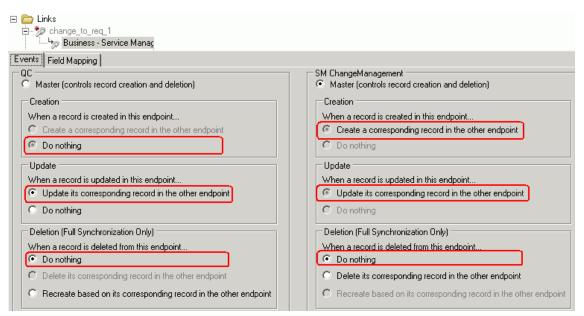
Note: Check for QCEntityID mapping property **Synchronize back on create**.

Define Events

The following table lists the event settings for the two endpoints.

Events Tab Settings	QC Action (Event)	SM Action (Event)
Creation	Do nothing.	Create a corresponding record in the other endpoint.
Update	Update its corresponding record in the other endpoint.	Update its corresponding record in the other endpoint.
Deletion	Do nothing.	Do nothing.

The following screenshot shows the settings:

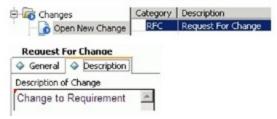


Test the Link

To test the link:

Note: The following is only 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 customization; RFC is used in this example).



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

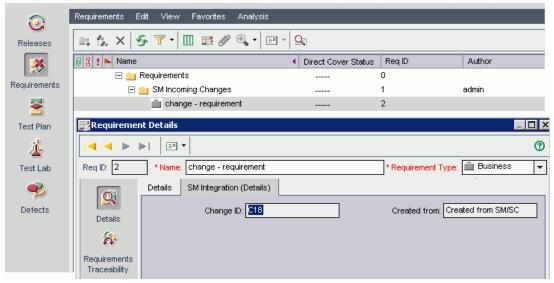
5. Select Forward as Requirement.

Backout Method	History	Approvals	Attachments	Related Records	Workflow	QC Integration	27
	Forward to	QC:		1 - Forward as Requ	irement	Ŧ	
	Def/Req ID						
		nain/Project:		ocalhost/DEFAULT/D	Anna A		

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.



Note: In ALM 11, the **Change ID** and **Created from** fields reside on the Details tab of the requirement form.

Chapter 8: SM Problem -> QC/ALM Defect

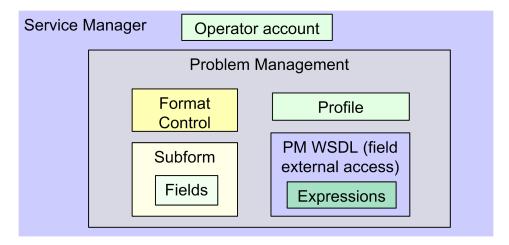
This chapter describes how to synchronize SM Problems with QC/ALM Defects ("Problem -> Defect").

This chapter includes:

- "Customizing Service Manager for Problem Management" below
- "Customizing the QC/ALM Defects Module" on page 112
- "Configuring Links in QC/ALM Synchronizer" on page 118

Customizing Service Manager for Problem Management

The following diagram summarizes the components which require tailoring in Service Manager.



To customize Service Manager for Problem Management, perform the following tasks:

- 1. "Add Fields" on the next page
- 2. "Specify the External Access Definition on Service Manager" on the next page
- 3. "Create a Subform" on page 101

- 4. "Add the Subform to a Form" on page 103
- 5. "Add Format Control Calculations/Validations" on page 105

Add Fields

Add the following required fields to the rootcause table. Do not change them.

	Туре				
Field	Service Manager	ServiceCenter			
qcintegration.type	Character	Text			
qcintegration.id	Number	Decimal			
qcintegration.project	Character	Text			

Note: The data type requirements for SM fields are described in "Matching Types" on page 31.

Specify the External Access Definition on Service Manager

If the Service Manager Process Designer (PD) Content Pack is not installed, or you have PD Content Pack 9.30.2 installed, follow the steps below to specify the External Access Definition on Service Manager:

Note: The following steps also apply to SM 9.4x Classic.

- Create a custom External Access Definition QCIntProblemService by clicking Tailoring > WSDL configuration on Service Manager 7.0x; or clicking Tailoring > Web Services > Web Service Configuration on Service Manager 7.1x or later with the following values:
 - Service Name: QCIntProblemService
 - Name: rootcause
 - Object Name: QCIntProblem
 - Allowed Actions / Action Names:

- add / Create
- save / Update

External Ac	External Access Definition							
Service Name:	QCIr	itProl	blemService					
Name:	rooto		•	Obje	ect Name:	QCIntProblem		
🗇 Allowed Actions 🛛 🧇		Expressions	٠	Fields				
Allowed Actions		Action Names			Action Typ	ре		
add		Create						
save		Update						
save		Update						

Note: The above values are required (Do NOT change them).

2. Enable the required fields in the web service.

Field	Caption	Туре
id	ProblemID	StringType
sysmodtime	Modified	DateTimeType
qcintegration.id	QCEntityID	IntType
qcintegration.project	QCProject	StringType
qcintegration.type	QCIntegrationType	StringType
qcintegration.created.from	CreatedFrom	StringType
current.phase	CurrentPhase	StringType
category	WorkFlowType	StringType
affected.itemsm.device.display.name	ServiceName	
logical.namesm.device.display.name	AffectedCIName	

E	kternal Acc	ess De	finition				
Se	ervice Name:	QCIntPr	roblemService		_		
Na	ame:	rootcau	se		=		
	Allowed Act	ions 🗇	Expressions	Expressions 🗇 Fields			
	Field		Caption		Туре		
1	qcintegration	QCEntityID		IntTy	pe		
	id	ProblemID		StringType			
	sysmodtime	Modified	DateTimeType				
U	qcintegration	QCProject	StringType				
	incident.cate	gory	Category		String	јТуре	
	subcategory		SubCategory	StringType			
	product.type	:	ProductType	StringType			
	problem.type	•	ProblemType	StringType			
	initial.impact		Impact		StringType		
	severity		Severity		StringType		
	description		Description		String	јТуре	
	assignment	AssignmentGr	oup	String	јТуре		
	ticket.owner		ProblemOwner	r	StringType		
	category		WorkFlowType	э	String	јТуре	

Note: The affected.item..sm.device.display.name field and the logical.name..sm.device.display.name are available on Service Manager 9.41 only. These fields are used for synchronizing SM Problem to QC/ALM Defects only.

Note: The caption value must be unique and alphanumerics (no spaces) with the first letter capitalized (AValidCaption123, AnotherValidCaption, and so on). The above values are required (Do NOT change them).

3. Define expressions for the web service.

Note:

- a. Problem Management requires an activity update provided with each save and for better flow, and this activity update will be hard-coded with the following expressions.
- b. Expressions 1 to 4 are for fixing an update issue with the Problem Management web service. For more information, see *SCR 41399*.

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	<pre>\$rc.update=update in \$L.file;if (denull(\$rc.update)={}) then (\$rc.update={"QC update sent"})</pre>
3	if (\$L.need.to.update=true) then (\$rc.update={"QC update sent"})
4	update in \$L.file=update in \$L.file.save

Service Name:	QCIntProblemService		
Name:	rootcause	✓ Object Name:	QCIntProblem
♦ Allowed Actions ♦ Expressions	Fields		
Expressions			
Expressions cleanup(\$pm.activity);cleanup	p(\$rc.update);if same(update in \$L.file, up	idate in \$L.file.save) then (\$L.need.to.update=tru	Je)
cleanup(\$pm.activity);cleanu	p(\$rc.update);if same(update in \$L.file, up ;if (denull(\$rc.update)={}) then (\$rc.upda		ie)
cleanup(\$pm.activity);cleanu \$rc.update=update in \$L.file;			je)
cleanup(\$pm.activity);cleanu \$rc.update=update in \$L.file;	if (denull(\$rc.update)={}) then (\$rc.upda then (\$rc.update={"QC update sent"})		е)
cleanup(\$pm.activity);cleanu \$rc.update=update in \$L.file; if (\$L.need.to.update=true) update in \$L.file=update in \$L	if (denull(\$rc.update)={}) then (\$rc.upda then (\$rc.update={"QC update sent"})		ie)

If the Service Manager Process Designer (PD) Content Pack 9.30.3 is installed, follow the steps below to specify the External Access Definition on Service Manager:

Note: The following steps also apply to SM 9.4x Codeless/Hybrid.

- 1. Refer to step 1.
- 2. Enable the required fields in the web service.

Field	Caption	Туре
id	ProblemID	StringType
sysmodtime	Modified	DateTimeType
qcintegration.id	QCEntityID	IntType
affected.itemsm.device.display.name	ServiceName	
logical.namesm.device.display.name	AffectedCIName	

External Access Definition

Service Name: Name: Object Name:		QCIntProblemService Tootcause QCIntProblem			Released: Deprecated:	
Allowed Actio	ons Expressions Fields					
-	Field	Caption		Турс		
id		Problem	۱D	StringType		
sy	/smodtime	Modifie	d	DateTimeType		
qc	cintegration.id	QCEntityID		IntType		
qcintegration.project qcintegration.type		QCProject		StringType		
		QCInteg	grationType	StringType		
qc	qcintegration.created.from Creat		from	StringType		

Note: The affected.item..sm.device.display.name field and the logical.name..sm.device.display.name are available on Service Manager 9.41 only.

Note: The caption value must be unique and alphanumerics (no spaces) with the first letter capitalized (AValidCaption123, AnotherValidCaption, and so on). The above values are required (Do NOT change them).

3. Define expressions for the web service.

Note: Problem Management requires an activity update provided with each save and for better flow, and this activity update will be hard-coded with the following expressions.

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	<pre>\$rc.update=update in \$L.file;if (denull(\$rc.update)={}) then (\$rc.update={"QC update sent"})</pre>
3	if (\$L.need.to.update=true) then (\$rc.update={"QC update sent"})
4	update in \$L.file=update in \$L.file.save
5	isKnownError in \$L.file=nullsub(isKnownError in \$L.file,true)

External Access Definition		
Service Name:	QCIntProblemService	Released
Name:	rootcause 👻	Deprecated
Object Name:	QCIntProblem	
Allowed Actions 🗇 Expressions	Fields	
<pre>\$rc.update=update in \$L.file;</pre>	p(\$rc.update);if same(update in \$L.file, update in \$L.file.sav if (denull(\$rc.update)={}) then (\$rc.update={"QC update ser nen (\$rc.update={"QC update sent"}) L.file.save ub(isKnownError in \$L.file.true)	

Note: You do not need to define the isKnownError in \$L.file=nullsub(isKnownError in \$L.file,true) expression on Service Manager 9.40 and 9.41.

Create a Subform

To create a subform:

- 1. Create a global list.
 - a. Create a global list by clicking Tailoring > Tailoring Tools > Global Lists on Service Manager
 7.0x or later; or clicking Utilities > Tools > Global Lists on ServiceCenter with the following parameters:

No.	Parameter	Value	Remarks
1	List Name	SMQC Integration PM Project List	
2	Regen Every	1 00:00:00	
3	Build List on Startup?	Yes	Check box
4	List Variable	\$G.qcintegration.problem.project	

No.	Parameter	Value	Remarks
5	User Defined List?	Yes	Check box
6	Value List	{"server1/domain1/project1", "server2/domain2/project2"}	Change to the values for your system
			Note: No spaces between slashes.

- b. Save this global list and click **Rebuild Global List** in the options menu.
- 2. Create a subform.

Click **Tailoring** > **Forms Designer** on Service Manager; or click **Toolkit** > **Forms Designer** on ServiceCenter to create the pm.qcint.subform subform with the following components:

Note: Click No when the system message "Do you want to use Form Wizard?" appears.

No.	Component	Properties
1	Label	Caption: Synchronize with QC:
2	Combo Box	 Input: qcintegration.type Value List: 0;1; Display List: 0 - Not Synchronize;1 - Synchronize with QC Defect Select Only: Yes
		Read-Only Condition: [\$qcint.type.readonly]
3	Label	Caption: Defect ID:
4	Text	 Input: qcintegration.id Read-Only: Yes
5	Label	Caption: Server/Domain/Project:

No.	Component	Properties	
6	Combo Box	Input: qcintegration.project	
		Value List: \$G.qcintegration.problem.project	
		Read-Only Condition: [\$qcint.project.readonly]	
		Mandatory Condition: [qcintegration.type]>0	

🗸 OK 😫 Cancel	
	10 🔏 🕂 🔛 📭 📰 🚥 🖬 🗸
Synchronize with QC:	•
Defect ID:	
Server/Domain/Project	▼
Created from:	

Add the Subform to a Form

If the Service Manager Process Designer (PD) Content Pack is not installed, or you have installed PD Content Pack 9.30.2, follow the steps below to add the subform you created to a form:

Note: The following steps also apply to SM 9.4x Classic.

- From Forms Designer, open the default form of a Problem Management phase (PM.pc.ident.and.class is used as an example on ServiceCenter 6.2/Service Manager 7.0x).
- 2. Add a notebook Tab with the QC Integration caption.
- 3. Add a subform to the new tab with format pm.qcint.subform, which is displayed in the following screenshot:

🗟 *Forms Designer: 🗙 🛃 Problem Managemen 🛛 📲 rootcausephase: 🎽 🔭 🗖	Properties 🛛	- 0
	Subfo	ormat
	Property	Value
III III 🖪 🚥 🧶 📓 🗮 🧮 🖉 🏷 🍢 📆 🖏 👗 🚺	Display Blank	\checkmark
	Display Using Table	
Problem Control - Problem Investigation and Diagnosis	Format	pm.qcint.subform
Record Number:	Height	40
	Input	
Expected Resolution E	Name	
Brief Description:	Virtual Join	
	Visible	
♦ SLA ♦ Related Records ♦ Tasks ♦ History ♦ Workflow ♦ QC Integration »s	Visible Condition	
	Width	143
	X	3
	Y	1

4. Save the changes.

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

If the Service Manager Process Designer (PD) Content Pack 9.30.3 is installed, follow the steps below to add the subform you created to a form:

Note: The following steps also apply to SM 9.4x Codeless/Hybrid.

- 1. From Forms Designer, open the default form of a Problem Management phase (pbm.problem.logging is used as an example on Service Manager 9.3x).
- 2. Add a notebook Tab with the QC Integration caption.
- 3. Add a subform to the new tab with format pbm.problem.logging, which is displayed in the following screenshot:

🔀 To Do Queue: My To Do List 🛛 📓 Forms Designer: pm.qcint.subfo 🛛 🔀 Forms Designer: pbm.problem.l 🕮 🗧	. 🗆	Properties 🛛	
	•	S	ubformat
III 💽 🗂 🗂 🗆 🖕 🍇 🖆 🔛 🛥 🚟 🖙 🖼 🖉 💿 🛒 🐂 🚏		Property	Value
🔟 🔟 📑 🖙 👄 🔜 🛲 🧮 🖉 🔣 🌄 🐯 🐯 🐯 🐻 崙 🛛		Name	
		X	5
		Y	2
		Width	72
		Height	22
		Visible	\checkmark
		Visible Condition	
♦ Affected Configur ♦ SLA ♦ Companies ♦ Change Calendar ♦ Attachments ♦ QC Integration »1		Tab Stop	0
		Format	pm.qcint.subform
		Virtual Join	
		Display Blank	\checkmark
		Display Using Table	
		Input	
•			
	-		
	-		

4. Save the changes.

Add Format Control Calculations/Validations

Note: Steps in this topic are applicable if the Service Manager Process Designer (PD) Content Pack is not installed, or if you have SM 9.3x with PD Content Pack 9.30.2 installed, or if you are using SM 9.4x Classic. Otherwise, skip this topic and refer to "Add Rule Set Calculations/Validations" on the next page for configurations in the SM 9.3x with PD Content Pack 9.30.3 and SM 9.4x Codeless/Hybrid environment.

To add format control calculations and validations, follow the steps below:

- 1. Open the format control record of the previous Problem Management form PM.pc.ident.and.class.
- 2. Click Calculations.
- 3. Add two rows with the following values:

Record	Parameter	Value
1	display	true
	initial	true
	calculation	<pre>\$qcint.type.readonly=2;if (qcintegration.type in \$file~=0) then (\$qcint.type.readonly=1)</pre>
2	display	true
	initial	true
	calculation	<pre>\$qcint.project.readonly=2;if (qcintegration.type in \$file~=0 and not null (qcintegration.project in \$file)) then (\$qcint.project.readonly=1)</pre>

4. Click Validations.

5. Add one row with the following values:

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

6. Save your changes.

Add Rule Set Calculations/Validations

Note: Steps in this topic are applicable for the Service Manager 9.3x with Process Designer (PD) Content Pack 9.30.3 and SM 9.4x Codeless/Hybrid only. Otherwise, refer to "Add Format Control Calculations/Validations" on the previous page for configurations in Service Manager 9.3x non-PD, 9.3x with PD 9.30.2, and SM 9.4x Classic environment.

To add rule set calculations and validations, follow the steps below:

- 1. "Copy an Existing Workflow" below
- 2. "Associate an Existing Problem Category with the New Workflow" below
- 3. "Create New Rule Set for Initialization and Validation" on the next page
- 4. "Associate the New Workflow with the New Rule Set" on page 110

Copy an Existing Workflow

You can use copies of the existing workflows in another business process, or make changes to the HP proprietary workflow copies.

To copy an existing workflow, follow the steps below:

- 1. From the System Navigator, click **Tailoring > Process Designer > Copy Existing Workflow**.
- 2. On the Clone a Workflow page, select the workflow you want to copy. For example, Problem.
- 3. Type SMQCIntPbM in the New workflow name field.

To Do	Queue: My To Do List	Wizard: Clone a Workflow 🗵	
CI	one a Workflo	N	
		Please specify the new workflow name, as well as the prefix for new rule sets if they are to be copied as v	well.
		New workflow name: * SMQCIntPbM	
		Copy rule sets?	

- 4. Select the Copy rule sets check box if you want to copy rule sets, and then type a rule set prefix.
- 5. Click **OK**.

The newly copied workflow appears in the list on the Clone a Workflow page.

Associate an Existing Problem Category with the New Workflow

You can update existing problem categories, subcategories and areas and associate the updated categories with the new workflow so that they can be used in another business process. For more information, refer to *HP Service Manager – Process Designer Content Pack Administrator's Guide*.

To associate an existing change category with the new workflow, follow the steps below:

- 1. From the System Navigator, click **Problem Management > Configuration > Problem Categories**.
- 2. Click Search.
- 3. Select the problem category for which you want to add a workflow. For example, problem.
- 4. In the Problem Category page, remove the currently assigned workflow from the Workflow field.
- 5. Type SMQCIntPbM in the workflow field.
- 6. Click **Save** to associate the problem category with the workflow.

Problem Category				
Name:	problem	Apply To:	Problem	
Active:				
Description:	incident			
Workflow:	* SMQCintPbM			
Subcategories Workflow				4
	orization -	olution \Rightarrow Review	Closure	

Create New Rule Set for Initialization and Validation

To create a new rule set for initialization, follow the steps below:

- 1. From the System Navigator, click **Tailoring > Process Designer > Rule Sets**.
- 2. Type the values as follow:

Field	Value
ID	pbm.alm.int.init
Available as action	False

Field	Value
Name	Initialize for ALM integration in the Problem Record
Table name	rootcause
HP Proprietary	

3. Click New and Save.

4. Click Add Rule.

- 5. In the Select Rule Type page, click Run JavaScript.
- 6. In the Run JavaScript page, type the values as follow:

Field	Value
Rule Description	Run Javascript for initializing Integration type and project in the Problem Record
Statement	<pre>vars['\$qcint.type.readonly'] = 2; vars['\$qcint.project.readonly'] = 2; var _null=system.functionsnull; var file = vars.\$L_file; if(file["qcintegration.type"] != 0 && !_null(file["qcintegration.type"])) { vars['\$qcint.type.readonly'] = 1 } if(file["qcintegration.type"] != 0 && !_null(file["qcintegration.project"])) {</pre>
	vars['\$qcint.project.readonly'] = 1 }

7. Click **OK**.

8. Click **Save** and **Exit**.

To create a new rule set for validation, follow the steps below:

1. From the System Navigator, click **Tailoring > Process Designer > Rule Sets**.

2. Type the values as follow:

Field	Value
ID	pbm.alm.int.validation
Available as action	False
Name	Validation for ALM integration in the Problem Record
Table name	rootcause
HP Proprietary	

- 3. Click New and Save.
- 4. Click Add Rule.
- 5. In the Select Rule Type page, click **Set Mandatory Fields**.
- 6. Refer to step 6 to step 13 as described in To create a new rule set for validation in "Create New Rule Set for Initialization and Validation" on page 60.

Associate the New Workflow with the New Rule Set

To associate the new workflow with the new rule set, follow the steps below:

- 1. From the System Navigator, click **Problem Management > Configuration > Workflows**.
- 2. Select SMQCIntPbM in the workflows list.
- 3. Select the first phase in the workflow graph.
- 4. Click **Rule Sets** tab > **Initialization** tab.
- 5. Click **Add** and select the pbm.alm.int.init rule set you just created.

MP Service Manager				User: falcon
	To Do Queue: My To Do List Workflows	Workflow: SMQCIntPbM (8)		
😨 🖸 😂 🛛 🔍 🔍	💾 Save 🔍 Zoom in 🔍 Zoom out 🗖 Add p	phase 🐨 Delete I 🚰 Workflow properties		8
Request Management Service Catalog Service Deak Service Level Management System Administration Tailoring > Audt			t¢ Course t→ Review	
Differential Upgrade Document Engine Event Services Knowledge Engineering Notifications	Abandonme	Add Rule Sets - Initialization	Name	
✓ Process Designer ▷ Configuration		apply.template create.template obm.abandon.wizard	Apply Template Create Template from Record	
Copy Existing Workflow Export Workflow Rule Sets	Phase - Logging Details Forms Rule Sets Actions	pbm.abandon.wizard pbm.abandonment obm.abandonment.field.validation	Run Abandon Wizard PBM abandonment PBM field validation at abandonment chase	
Workflows E ▷ SQL Utilities ▷ Tailoring Tools ▷ Web Services	On enter On exit (Initialization) O ♣Add ⑦ Delete ↔ View ♠ Up ↓ Down	phm aim int validation	Initialize for ALM integration in the Problem Record Validation for ALM integration in the Problem Record Associate with BAC PI	
Codes Database Dictionary	Rule Sets	pbm.calculate.rc.calendar pbm.categorization.init.status	Calculate whether RC calendar needs to be displayed Initialize pm status at categorization phase	
Database Manager Data Policy Format Control			OK Cancel	

- 6. Click **OK**.
- 7. Click **Rule Sets** tab > **On display** tab.
- 8. Repeat step 5 and 6.
- 9. Click **Rule Sets** tab > **On enter** tab.
- 10. Click **Add** and select the pbm.alm.int.validation rule set you just created.

MP Service Manager					
		To Do Queue: My To Do List Workflows	Workflow: SMQCIntPbM 🗵		
12 2 3	~	💾 Save 🍕 Zoom in 🤤 Zoom out 🗖 A	Add phase 🐨 Delete 🚰 Workflow properties		
- Request Management	^				
Service Catalog					
Service Desk					
Service Level Management			\$	¢ \$	
System Administration					
		Logging Categoriza	ation 🗘 Nrvestigation ⊅	Resolution 🗘 Review Closure	
Tailoring	_	a			
Audit					
 Differential Upgrade Document Engine 					
 Document Engine Event Services 		Abando	Add Rule Sets - On enter		
 Event Services Knowledge Engineering 					
 Notifications 			ld Id	Name	
Process Designer			apply.template	Apply Template	-
 Process Designer Configuration 			create.template	Create Template from Record	=
Copy Existing Workflow			pbm.abandon.wizard	Run Abandon Wizard	
Export Workflow		Phase - Logging	pbm.abandonment	PBM abandonment	
Rule Sets		Details Forms Rule Sets Actions		PBM field validation at abandonment phase	
Workflows	E		bin abandoninen: neid vandation		
SQL Utilities		On enter On exit Initialization	On (Contraction of the second	Initialize for ALM integration in the Problem Record	_
Tailoring Tools		🖨 Add 💮 Delete 😚 View 🏫 Up 🐺 D	pbm.alm.int.validation	Validation for ALM integration in the Problem Record	
> Web Services			pbm.bac.pi.assciate	Associate with BAC PI	
Codes		Rule Sets	pbm.calculate.rc.calendar	Calculate whether RC calendar needs to be displayed	
Database Dictionary			pbm.categorization.init.status	Initialize pm status at categorization phase	
Database Manager					
Data Policy				OK Can	cel
Format Control					
Forms Designer					
Integration Manager					
Localization Utility					

- 11. Click **OK**.
- 12. Click Save.

Customizing the QC/ALM Defects Module

The steps for customizing the Defects module vary with different QC versions.

- "On QC 10 and Earlier" below
- "On ALM 11" on page 116

On QC 10 and Earlier

To customize the Defects module on Quality Center 10 or earlier, perform the following tasks:

- 1. "Add Fields" below
- 2. "Add Tabs" on the next page
- 3. "Add Fields to Tabs" on page 114

Add Fields

To add the required fields for Defect module customization:

- 1. Log on to QC as a project administrator.
- 2. Click **Tools / Customize**. The "QC Project Customization" module opens.
- 3. Add the following fields for the defect entity in Project Entities (*XX* and *XY* are sequential numbers auto-generated by QC).

Field Name	Field Label	Field Type	Remarks
BG_USER_XX	Problem ID	String	
BG_USER_XY	Created from	String	

The following figure shows an example project entity.

Quality Center - I	Project Customization		
User Properties Project Users Groups Module Access Project Entities Requirement Types Risk-Based Quality Project Lists Automail Alert Rules	Project Customization Project Customization Project Entities Project Entities Defect User Fields User Fields User Fields Forward as problem Rec Problem ID	Field Settings Field Name: Field Label: Field Type: Field Length:	BG_USER_01) Forward as problem Lookup List
Workflow	1 :	History	🗌 Required
		Masked	Searchable

Note: The data type requirements for QC fields are described in "Matching Types" on page 31.

Add Tabs

To add tabs to the Defect form and show fields on these tabs:

1. In "QC - Project Customization", click **Workflow** > **Script Editor**.

2. Select Defects module script.

Quality Center - Project Customization

User Properti Project Users		low
Groups Module Acce Project Entitie Requirement Risk-Based G Project Lists Automail	SE Enables S You can Types Script G huality Man Enables	enerator - Add Defect Field Customization s you to customize the fields displayed for each user group in the Add Defects dialog box also specify field order and whether a field is required. enerator - Defect Details Field Customization you to customize the fields displayed for each user group in the Defect Details dialog c an also specify field order and whether a field is required.
Alert Rules Workflow	or You can	s you to write VBScript code for all Quality Center modules. also use the Script Editor to modify the scripts generated by the above tools.
		(Cethewart reparenterne)
Script Editor	Toolbar Button Editor	SetNewBugPageName

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

Note: The parameter 2 specifies tab 2 (the second tab). For a new bug, the tab name is SM Integration (New).

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

select case PageNum

case "2"

```
GetDetailsPageName = "SM Integration (Details)"
```

end select

Note: The parameter 2 specifies tab 2 (the second tab). For an existing defect, the tab name is SM Integration (Details).

Add Fields to Tabs

To add fields to tabs:

1. In "QC - Project Customization", click **Workflow** > **Script Editor**.

2. Select Defects module script.

E- Defects module script

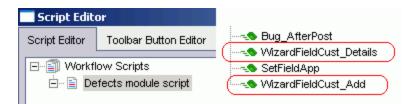
Quality Center - Project Customization

User Properties Project Users	Workflow
Groups Module Access Project Entities Requirement Types Risk-Based Guilty Man Project Lists Automal	Script Generator - Add Defect Field Customization Enables you to customize the fields displayed for each user group in the Add Defects dialog box You can also specify field order and whether a field is required. Script Generator - Defect Details Field Customization Enables you to customize the fields displayed for each user group in the Defect Details dialog box. You can also specify field order and whether a field is required.
Alert Rules S Workflow	Script Editor Enables you to write VBScript code for all Quality Center modules. You can also use the Script Editor to modify the scripts generated by the above tools.
Script Editor Script Editor Toolbar Bu	tion Editor GetNewBugPageName
Workflow Scripts Ommon script Script Common script Script Common script Script Test Plan modu Script Test Lab modul	nodule script Sug_FieldConChange

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

Bug_AfterPost SetFieldApp

- a. Script Generator Add Defect Field Customization
- b. Script Generator Defect Details Field Customization



4. 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

The parameter values are:

Installation and Administration Guide Chapter 8: SM Problem -> QC/ALM Defect

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

6. Set the **Readonly** fields by adding the following lines to the **Bug_New** and **Bug_Moveto** subroutines:

Bug_Fields.Field("BG_USER_XX").IsReadOnly=True

Bug_Fields.Field("BG_USER_XY").IsReadOnly=True

7. Save your changes.

On ALM 11

On ALM 11, you only need to add new fields directly to the Details tab of the Defect form.

To customize the ALM Defects module, perform the following task:

"Add Fields" below

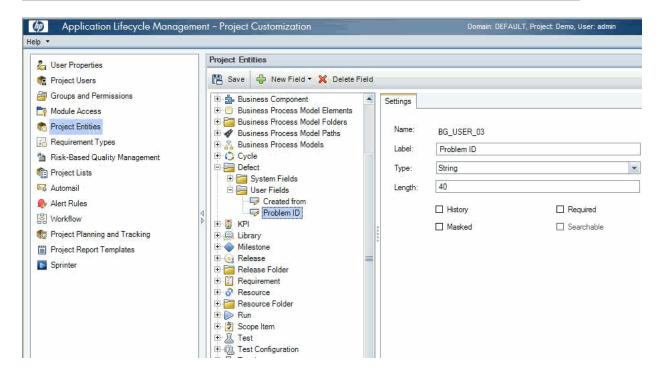
Add Fields

To add the required fields for Defect module customization:

- 1. Log on to ALM as a project administrator.
- Click Tools / Customize. The "Application Lifecycle Management Project Customization" module opens.
- 3. Add the following fields for the defect entity in Project Entities (XX and XY are sequential numbers

auto-generated by ALM).

Field Name	Field Label	Field Type
BG_USER_XX	Problem ID	String
BG_USER_XY	Created from	String



Note: The data type requirements for QC/ALM fields are described in "Matching Types" on page 31.

- 4. Click Workflow > Script Editor.
- 5. Select Defects module script.
- Set the **Readonly** fields by adding the following lines to the **Bug_New** and **Bug_Moveto** subroutines:

Bug_Fields.Field("BG_USER_XY").IsReadOnly=True

Bug_Fields.Field("BG_USER_XZ").IsReadOnly=True

7. Save your changes.

Configuring Links in QC/ALM Synchronizer

To configure and test a link in the QC/ALM synchronizer, perform the following tasks:

- 1. "Specify Endpoints / Type of Link" below
- 2. "Define Field Mappings" below
- 3. "Define Events" on the next page
- 4. "Test the Link" on page 120

Specify Endpoints / Type of Link

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

- 1. Step 1: "Endpoint 2 type" = SM ProblemManagement.
- Step 3: "Service URL" = http://service_manager_ host>:<port>/sc62server/PWS/QCIntProblemService.wsdl
- 3. Step 4: "Select entity types" = Problem by Defect (this is the only available selection).

Define Field Mappings

If the Service Manager Process Designer (PD) Content Pack is not installed, or if you are working with SM 9.4x Classic, see the following summary for basic field mappings:

QC	Direction	SM	Constant value	Remarks
Problem ID	<-	ProblemNumber		
Defect ID	->	QCEntityID		Synchronize back on create: Yes
Created from			Created from SM/SC	

Example field mappings are shown in the following screenshot:

Mappe	Mapped Fields				
Туре	QC Field	Direction	SM ChangeManagement Field		
	Severity	<>	Urgency		
	Problem ID	<	ProblemI D		
	Defect ID	>	QCEntityID		
	Summary	<>	Description		
$\langle \rangle$	Created from	<	Value: Created from SM/SC		

If the Service Manager Process Designer (PD) Content Pack is installed, or if you are working with SM 9.4x Codeless/Hybrid, status drive the business process of Help Desk in Process Designer Content Pack 9.30.3 and SM 9.4x Codeless/Hybrid. You need to pay attention to the value mapping about the **Status** field between Service Manager and ALM.

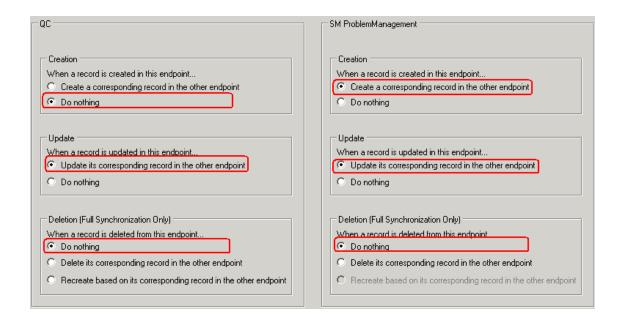
Define Events

The following table lists the event settings for the two endpoints:

Events Tab Settings	QC Action (Event)	SM Action (Event)
Creation	Do nothing.	Create a corresponding record in the other endpoint.
Update	Update its corresponding record in the other endpoint.	Update its corresponding record in the other endpoint.
Deletion	Do nothing.	Do nothing.

The following screenshot displays the settings:

Installation and Administration Guide Chapter 8: SM Problem -> QC/ALM Defect



Test the Link

To test the link:

Note:

- 1. 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.
- 2. The following sample steps are for your reference only. The exact steps required on your system may differ significantly. The phase in which the QC Integration tab appears may be different on your system.
- 1. Save the configuration (an integrity check is automatically run).
- 2. Click Enable Link.

3. Create a Problem and select **Synchronize with QC Defect**.

G	🗓 Problem PM0016 has been	opened.		(p)
	Problem Control - Proble	m Identification and Classification		
	Record Number:	PM0016 Status: Open Expected Resolution Date:	•	
	Brief Description:	problem to defect		
	Classification	♦ Attachments ♦ Related Records ♦ History ♦ Workflow ♦ QC Integration		
	Synchronize with Q(1 - Synchronize with QC Defect		
	Defect ID:			
	Server/Domain/Proj	localhost/DEFAULT/Demo		

4. Synchronize.

🛞 Cancel Current Task 📄 View Report	😯 Refresh Progress 👿 Auto Refresh
Running: Querying non-filtered set Running: Handling endpoint 1 - Processing Passed: Disconnecting Completed : Passed	entity #1 of #1 in the Create list, (Total: passed = 0, failed = 0)

5. View the Problem in SM.

								(P)		$\langle p \rangle$
Problem Cont	rol - Probler	n Identifi	cation ar	nd Classificat	ion					
Record Number:		PM00	016		Status: Expected	Resolutio	n Date:	Updated	 • •	
Brief Description:		prob	lem to defe	ct				L		
Classification	Activities	🔶 Attachm	ents 🧇 R	Related Records	🔷 History 🛛 🔷 V	Vorkflow	🔷 QC Integrat	ion		
Sync	hronize with QC	:		1 - 5	ynchronize with QC	Defect	~			
Defe	ct ID:			21						
Servi	er/Domain/Proje	ect		local	host/DEFAULT/Dem	10	~			

6. View the defect in QC.

Note: In ALM 11, the Problem ID field and the Created from field reside on the Details tab of the Defect form.

Chapter 9: QC/ALM Defect -> SM Problem

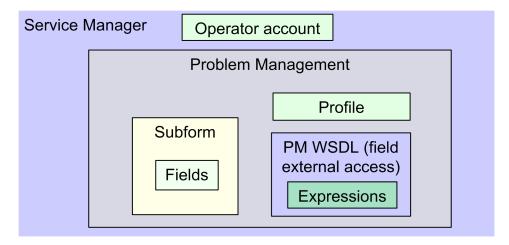
This chapter describes how to synchronize QC/ALM Defects with SM Problems ("Defect -> Problem ").

This chapter includes:

- "Customizing Service Manager for Problem Management" below
- "Customizing the QC/ALM Defects Module" on page 130
- "Configuring Links in QC/ALM Synchronizer" on page 139

Customizing Service Manager for Problem Management

The following diagram summarizes the components which require tailoring in Service Manager.



To customize Service Manager for Problem Management, perform the following tasks:

- 1. "Add Fields" on the next page
- 2. "Specify the External Access Definition on Service Manager" on the next page
- 3. "Create a Subform" on page 127
- 4. "Add the Subform to a Form" on page 128

Add Fields

Add the following required fields to the rootcause table. Do not change them.

	Туре			
Field	Service Manager 7.0x or later	ServiceCenter		
qcintegration.type	Character	Text		
qcintegration.id	Number	Decimal		
qcintegration.project	Character	Text		
qcintegration.created.from	Character	Text		

Note: The data type requirements for SM fields are described in "Matching Types" on page 31.

Specify the External Access Definition on Service Manager

To specify the External Access Definition on Service Manager:

- Create a custom External Access Definition QCIntProblemService by clicking Tailoring > WSDL configuration on Service Manager 7.0x; or clicking Tailoring > Web Services > Web Service Configuration on Service Manager 7.1x or later with the following values:
 - Service Name: QCIntProblemService
 - Name: rootcause
 - Object Name: QCIntProblem
 - Allowed Actions / Action Names:
 - add / Create
 - save / Update

External Access Definition									
Service Name:	QCIntPro	blemService							
Name:	rootcause	•	 Obje 	ect Name:	QCIntProblem				
Allowed Act	tions 🔷	Expressions	🔷 Fields						
Allowed Actions		Action Name	es	Action Typ	ре				
add		Create							
save		Update							

Note: The above values are required (Do NOT change them).

2. Enable the required fields in the web service.

Field	Caption	Туре
id	ProblemID	StringType
sysmodtime	Modified	DateTimeType
qcintegration.id	QCEntityID	IntType
qcintegration.project	QCProject	StringType
qcintegration.type	QCIntegrationType	StringType
qcintegration.created.from	CreatedFrom	StringType
current.phase	CurrentPhase	StringType
category	WorkFlowType	StringType

External Access	External Access Definition							
Service Name: QCIntProblemService								
Name: roote	cau	se		=				
Allowed Actions	4	Expressions	🔶 F	ields				
Field		Caption		Туре				
qcintegration.id		QCEntityID		IntType				
id		ProblemID		StringType				
sysmodtime		Modified		DateTimeType				
qcintegration.proje	ect	QCProject		StringType				
incident.category		Category		StringType				
subcategory		SubCategory		StringType				
product.type		ProductType		StringType				
problem.type		ProblemType		StringType				
initial.impact		Impact		StringType				
severity		Severity		StringType				
description		Description		StringType				
assignment		AssignmentGro	oup	StringType				
ticket.owner	ticket.owner			StringType				
category		WorkFlowType	•	StringType				

Note: To create known errors on Service Manager 9.40 and 9.41, you also need to add the following additional field to the required fields list:

Field	Caption	Туре
root.cause	RootCause	array of string

Note: The caption value must be unique and alphanumerics (no spaces) with the first letter capitalized (AValidCaption123, AnotherValidCaption, and so on). The above values are required (Do NOT change them).

3. Define expressions for the web service .

Note:

a. Problem Management requires an activity update provided with each save and for better flow, and this activity update will be hard-coded with the following expressions.

b. Expressions 1 to 4 are for fixing an update issue with the Problem Management web service. For more information, see *SCR 41399*.

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	<pre>\$rc.update=update in \$L.file;if (denull(\$rc.update)={}) then (\$rc.update={"QC update sent"})</pre>
3	if (\$L.need.to.update=true) then (\$rc.update={"QC update sent"})
4	update in \$L.file=update in \$L.file.save

Service Name:	QCIntProblemService			
Jame:	rootcause	•	Object Name:	QCIntProbler
♦ Allowed Actions	Fields			
,				
Expressions				
LAPI 03310113				
)(\$rc.update);if same(update in \$L.file, u	pdate in \$L.file.save) then (\$L.need.to.update=true)	
cleanup(\$pm.activity);cleanup)(\$rc.update);if same(update in \$L.file, u if (denull(\$rc.update)={}) then (\$rc.upda		\$L.need.to.update=true)	
cleanup(\$pm.activity);cleanup \$rc.update=update in \$L.file;i			\$L.need.to.update=true)	
cleanup(\$pm.activity);cleanup \$rc.update=update in \$L.file;i	f (denull(\$rc.update)={}) then (\$rc.upda hen (\$rc.update={"QC update sent"})		\$L.need.to.update=true)	
cleanup(\$pm.activity);cleanup \$rc.update=update in \$L.file; if (\$L.need.to.update=true) t update in \$L.file=update in \$L	f (denull(\$rc.update)={}) then (\$rc.upda hen (\$rc.update={"QC update sent"})		\$L.need.to.update=true)	

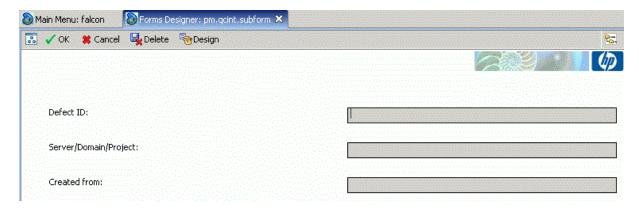
Create a Subform

To create a subform:

Click **Tailoring > Forms Designer** on Service Manager7.0x or later; or click **Toolkit > Forms Designer** on ServiceCenter to create the pm.qcint.subform subform with the following components:

Note: Click No when the system message "Do you want to use Form Wizard?" appears.

No.	Component	Properties
1	Label	Caption: Defect ID:
2	Text	• Input: qcintegration.id
		• Read-Only: Yes
3	Label	Caption: Server/Domain/Project:
4	Text	Input: qcintegration.project
		• Read-Only: Yes
5	Label	Caption: Created from:
6	Text	Input: qcintegration.created.from
		• Read-Only: Yes



Add the Subform to a Form

If the Service Manager Process Designer (PD) Content Pack is not installed, or you have PD Content Pack 9.30.2 installed, follow the steps below to add the subform you created to a form:

Note: The following steps also apply to SM 9.4x Classic.

- From Forms Designer, open the default form of a Problem Management phase (PM.pc.ident.and.class is used as an example in ServiceCenter 6.2/Service Manager 7.0x).
- 2. Add a notebook tab with the QC Integration caption.
- 3. Add a subform to the new tab with format pm.qcint.subform, which is displayed in the following screenshot:

🔓 *Forms Designer: 🗙 👺 Problem Managemen 🛛 👺 rootcausephase: 🎽 👘	l 🔲 Properties 🛛	- 8
	Subfo	ormat
	Property	Value
🔟 🔟 📑 🖙 👁 🔜 🛲 🧮 🖉 🏈 🌄 📆 🎇 👫 🚺	Display Blank	
	Display Using Table	
Problem Control - Problem Investigation and Diagnosis	Format	pm.qcint.subform
Desard Musham	Height	40
Record Number:	Input	
Expected Resolution E	Name	
Brief Description:	Virtual Join	
	Visible	
♦ SLA ♦ Related Records ♦ Tasks ♦ History ♦ Workflow ♦ OC Integration »5	Visible Condition	
	Width	143
	X	3
	Y	1

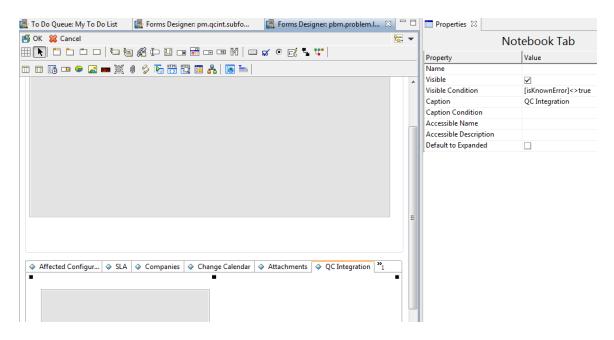
4. Save the changes.

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

If the Service Manager Process Designer (PD) Content Pack 9.30.3 is installed, follow the steps below to add the subform you created to a form:

Note: The following steps also apply to SM 9.4x Codeless/Hybrid.

- From Forms Designer, open the default form of a Problem Management phase (pbm.problem.logging is used as an example on Service Manager 9.3x) or open the default form of a Known Error phase (pbm.knownerror.logging is used as an example on Service Manager 9.40 Codeless and 9.41 Codeless/Hybrid).
- 2. Add a Notebook tab with the QC Integration caption and then set the value of Visible Condition to [isKnownError]<>true.



Note: You do not need to add Visible Condition to the new Notebook tab on Service Manager 9.40 and 9.41.

3. Add a subform to the new tab with format pbm.problem.logging, which is displayed in the following screenshot:

🛃 To Do Queue: My To Do List 🛛 📓 Forms Designer: pm.qcint.subfo 🛛 📓 Forms Designer: pbm.problem.l 🖂		Properties 🛛	
	-		Subformat
		Property	Value
III III 🔀 🚥 📾 📓 🌒 🔗 🍢 🐻 🐯 🖾 🖬 🔧		Name	
		X	5
		Y	2
		Width	72
		Height	22
		Visible	\checkmark
	J	Visible Condition	
♦ Affected Configur ♦ SLA ♦ Companies ♦ Change Calendar ♦ Attachments ♦ QC Integration »1		Tab Stop	0
		Format	pm.qcint.subform
		Virtual Join	
		Display Blank	\checkmark
		Display Using Table	
		Input	
	E		
· · · ·			

4. Save the changes.

Customizing the QC/ALM Defects Module

The steps for customizing the Defects module vary with different QC versions.

- "On QC 10 and Earlier" below
- "On ALM 11" on page 135

On QC 10 and Earlier

To customize the Defects module on Quality Center 10 or earlier, perform the following tasks:

- 1. "Add Fields" on the next page
- 2. "Add Tabs" on the next page
- 3. "Add Fields to Tabs" on page 133
- 4. "Create a View" on page 135
- 5. "Verify" on page 135

Add Fields

To add the required fields for Defect module customization:

- 1. Log on to QC as a project administrator.
- 2. Click **Tools / Customize**. The "QC Project Customization" module opens.
- 3. Add the following fields for the defect entity in Project Entities (*XX* and *XY* are sequential numbers auto-generated by QC).

Field Name	Field Label	Field Type	Remarks
BG_USER_ <i>XX</i>	Synchronize with SM Problem	Lookup List/YesNo	Select the "Verify Value" check box
BG_USER_ <i>XY</i>	Problem ID	String	

Quality Center - Project Customization

<u>User Properties</u> Project Users	Project Entities	
Groups Module Access Project Entities Requirement Types Risk-Based Quality Project Lists Automail Alert Rules Workflow	Project Entities	Field Settings Field Name: BG_USER_01 Field Label: Forward as problem Field Type: Lookup List Field Length: 40 History Required Masked Searchable

Note: The data type requirements for QC fields are described in "Matching Types" on page 31.

Add Tabs

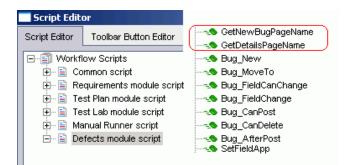
To add tabs to the Defect form and show fields on these tabs:

1. In "QC - Project Customization", click **Workflow** > **Script Editor**.

Quality Center - Project Customization

User Properties Project Users	Workflow
Groups Module Access Project Entities Requirement Types Risk-Based Quality Man Project Lists Automail	Script Generator - Add Defect Field Customization Enables you to customize the fields displayed for each user group in the Add Defects dialog box. You can also specify field order and whether a field is required. Script Generator - Defect Details Field Customization Enables you to customize the fields displayed for each user group in the Defect Details Field Customization Enables you to customize the fields displayed for each user group in the Defect Details dialog box. You can also specify field order and whether a field is required.
<u>Alert Rules</u> Workflow	Script Editor Enables you to write VBScript code for all Quality Center modules. You can also use the Script Editor to modify the scripts generated by the above tools.

2. Select Defects module script.



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

Note: The parameter 2 specifies tab 2 (the second tab). For a new bug, the tab name is SM Integration (New).

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

select case PageNum

case "2"

GetDetailsPageName = "SM Integration (Details)"

end select

Note: The parameter 2 specifies tab 2 (the second tab). For an existing defect, the tab name is SM Integration (Details).

Add Fields to Tabs

To add fields to tabs:

- 1. In "QC Project Customization", click **Workflow** > **Script Editor**.
- 2. Select Defects module script.

Quality Center - Project Customization

User Properties Project Users	Vorkflow
Groups Module Access Project Entities Requirement Types Risk-Based Quality Man Project Lists	Script Generator - Add Defect Field Customization Enables you to customize the fields displayed for each user group in the Add Defects dialog bo You can also specify field order and whether a field is required. Script Generator - Defect Details Field Customization Enables you to customize the fields displayed for each user group in the Defect Details dialog box. You can also specify field order and whether a field is required.
Workflow	<u>Script Editor</u> Enables you to write VBScript code for all Quality Center modules. You can also use the Script Editor to modify the scripts generated by the above tools.
Script Editor Toolbar Button	Editor GetNewBugPageName

- 3. If **WizardFieldCust_Details** and **WizardFieldCust_Add** are not found in the list, do the following to generate these two methods.
 - a. Script Generator Add Defect Field Customization
 - b. Script Generator Defect Details FieldCustomization



4. 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

The parameters are:

- Field name (BG_USER_XX, where XX consists of two digits)
- Visible (True)
- Required (False)
- Page number (start from 0)
- View order (start from 0)
- 5. 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

6. Set the **Readonly** fields by adding the following lines to the **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

The if loop above marks the field "Synchronize with SM Problem" as read-only after selected and saved.

7. Save your changes.

Create a View

To create a view:

- 1. Log on to QC with the integration account SMQCIntUser.
- 2. In the Defects module, click **View / Filter/Sort / Set Filters/Sort**. The purpose of this view is to make the QC Synchronizer correctly filter those defects to be synchronized to SM as Problems.
- 3. Set Synchronize with SM Problem to Y.
- 4. Add a view to Favorites:
 - Name: SMIntegrationView
 - Location: Private

Add Favoril	te	×
Add Favori	te	
Name:	SMIntegration∀iew	
Location:	Private O Public	;

In QC Synchronizer this view will be selected as the QC data filter. Without this filter, QC defects cannot be forwarded to SM as Problems.

Verify

Refer to the following screenshot to verify whether the Defects module on Quality Center 10 or earlier is customized successfully:

Details	SM Integration (Details)			
Syncl	hronize with SM Problem:		Problem ID:	PM0017
	Created from:	Created from SM/SC		

On ALM 11

On ALM 11, you only need to add new fields directly to the Details tab of the Defect form.

To customize the ALM Defects module, perform the following task:

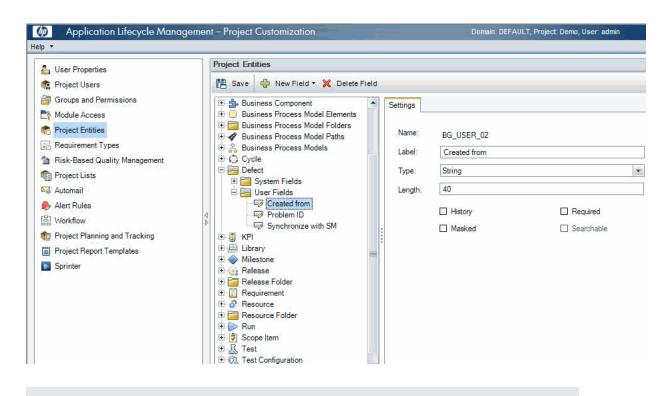
- 1. "Add Fields" below
- 2. "Create a View" on page 138
- 3. "Verify" on the previous page

Add Fields

To add the required fields for Defect customization:

- 1. Log on to ALM as a project administrator.
- 2. Click **Tools / Customize**. The "Application Lifecycle Management Project Customization" module opens.
- 3. Add the following fields for the defect entity in Project Entities (*XX* and *XY* are sequential numbers auto-generated by ALM).

Field Name	Field Label	Field Type	Remarks
BG_USER_ <i>XX</i>	Synchronize with SM Problem	Lookup List/YesNo	Select the "Verify Value" check box
BG_USER_ <i>XY</i>	Problem ID	String	



Note: The data type requirements for QC/ALM fields are described in "Matching Types" on page 31.

- 4. Click Workflow > Script Editor.
- 5. Choose Defects module script.
- Set the **Readonly** fields by adding the following lines to the **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

The if loop above marks the field "Synchronize with SM Problem" as read-only after selected and saved.

7. Save your changes.

Create a View

To create a view:

- 1. Log on to ALM with the integration account SMQCIntUser.
- 2. In the Defects module, click **View / Filter/Sort / Set Filters/Sort**. The purpose of this view is to make ALM Synchronizer correctly filter those defects to be synchronized to SM as problems.
- 3. Set Synchronize with SM Problem to Y.
- 4. Add a view to Favorites:
 - Name: SMIntegrationView
 - Location: Private

Add Favori	te			×
Add Favori	te			
Name:	SMIntegration∀iew			
Location:	 Private 	0	Public	

In ALM Synchronizer this view will be selected as the ALM data filter. Without this filter, ALM defects cannot be forwarded to SM as Problems.

Verify

Open a new defect in ALM, select **Y** in the **Synchronize with SM** field, and click **Save**. If the Defects module on ALM 11 is customized successfully, the Defect form is displayed as follow:

Details					
Actual Fix Time:			Synchronize with SM:	Y	
Closing Date:		•	Closed in Version:		•
* Detected By:	SMQCIntUser		Detected in Cycle:		•
Detected in Release:		-	* Detected on Date:	2010-11-12	•
Detected in Version:		-	Estimated Fix Time:		
Planned Closing Ver		•	Priority:		-
Project:		•	Reproducible:	Y	-
Assigned To:		•	* Severity:	2-Medium	-
Status:	New	•	Subject:		-
Target Cycle:		•	Target Release:		-
Change ID:			Created from:		
Problem ID:			Modified:		

Configuring Links in QC/ALM Synchronizer

To configure and test a link in the QC/ALM synchronizer, perform the following tasks:

- "Specify Endpoints / Type of Link" below
- "Define Filters" on the next page
- "Define Field Mappings" on the next page
- "Define Events" on page 146
- "Test the Link" on page 146

Specify Endpoints / Type of Link

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

- 1. Step 1: "Endpoint 2 type" = SM ProblemManagement.
- 2. Step 2 "Configuration File Name" = configuration_file_default.xml

- 3. Step 3: "Service URL" = http://service_manager_ host>:<port>/sc62server/PWS/QCIntProblemService.wsdl
- 4. Step 4: "Select entity types" = **Problem by Defect** (this is the only available selection).

Define Filters

On the Filters tab, select filter **SMIntegrationView** for the QC endpoint. If the filter is not available, see "Create a View" on page 135.

	SM ProblemManagement
C No Filter	No Filter
 Use filter (for creation events): 	C Use filter (for creation events):
Private: SMIntegrationView	

Define Field Mappings

Basic field mappings are summarized below:

QC	Direction	SM	Constant Value	Remarks
Problem ID	<-	ProblemID		
Defect ID	->	QCEntityID		Synchronize back on create: Yes
		QCIntegrationType	1	
		CreatedFrom	Created from QC	
QCProject	<-		(your setup)	This constant value should be the same as that for the "QC Project" parameter in the Connectivity tab.
	->	CurrentPhase	XXX	Replace XXX with a valid phase name, such as "Problem Investigation and Diagnosis". This field mapping is optional for Service Manager 7.10.

QC	Direction	SM	Constant Value	Remarks
	->	WorkFlowType	үүү	Replace YYY with a valid category name, such as <i>ITIL</i> for demo data of SM 7.0x/SC 6.2; <i>BPPM</i> is for demo data of Service Manager 7.10. This field mapping is optional for Service Manager 7.10.

Sample field mappings between Service Manager 7.10 and QC 10 are shown in the following screenshot:

Mapped Fields							
Туре	QC Field	Direction	SM ProblemManagement Field				
	Summary	<>	Description				
	Defect ID	>	QCEntityID				
- C	Severity	<>	Severity				
- Ci	Problem ID	<	ProblemID				
8	Value: Created from Quality C	>	CreatedFrom				
- 🐼	Value: 1	>	QCIntegrationType				
8	Value: AUTO	>	AssignmentGroup				
- 🐼	Value: BOB.HELPDESK	>	ProblemOwner				
8	Value: client system	>	Category				
<₿	Value: software	>	SubCategory				
<	Value: email client	>	ProductType				
- 🐼	Value: outlook	>	ProblemType				
8	Value: 4 - User	>	Impact				
<	Value: Problem Identification	>	CurrentPhase				
8	Value: localhost/DEFAULT/	>	QCProject				
<∂	Value: ITIL	>	WorkFlowType				

Sample field mappings between Service Manager 9.20 or later and ALM 11 are shown in the following screenshot:

Installation and Administration Guide Chapter 9: QC/ALM Defect -> SM Problem

Mapped Fields							
Туре	HP-ALM Field	Direction	SM-ProblemManagement Field				
	ProblemID	<	ProblemID				
	Summary	<>	Description				
- C	Severity	<>	Severity				
	Defect ID	>	QCEntityID				
8	CreatedFrom	<	Value: SM				
8	Detected on Date	<	Value: 11/2/2010				
<	Value: QC	>	CreatedFrom				
8	Value: 1	>	QCIntegrationType				
<₿	Value: 4 - User	>	Impact				
<	Value: DummyQCServer/DEF	>	QCProject				
$\langle \rangle$	Value: Application	>	AssignmentGroup				
<	Value: Software	>	AffectedItem				
$\langle \rangle$	Value: Problem Detection, Lo	>	CurrentPhase				
$\langle \rangle$	Value: problem	>	Category				
<	Value: hardware	>	SubCategory				
- 🔗	Value: hardware failure	>	ProductType				

QC/ALM Field <-> SM Field

If your Service Manager 9.3x has no Process Designer (PD) Content Pack installed, or if your are working with SM 9.4x Classic, see the following summary for basic field mappings between QC/ALM and SM. The first two rows are required mappings.

QC Len	QC DB Name	QC Type	QC/ QCS Label	QC/ QCS Type	Dir	QCS SM Type	QCS Name/ SM WSDL Caption	SM WSDL Type	SM DB Name/ SM WSDL field	SM7 DB type	SC6 DB type	SM Len
40	BG_USER _03	String	Problem ID	String	<-	String	Problem ID	StringType	id	Char	Text	100
10	BG_BUG _ID	Number	Defect ID	Number	->	Number	QCEntityID	IntType	qcintegration.id	Num	Decimal	хх
255	BG_SUMMARY	String	Summary	String	<->	String	Description	StringType	description	Char	Text	хх
70	BG_SEVERITY	Lookup List	Severity	Single value list	<->	Single value list	Severity		severity	Char	Text	40

QC Defect -> SM Problem Mappings

If your Service Manager has Process Designer (PD) Content Pack is installed, or if your are working with SM 9.4x Codeless/Hybrid, status drive the business process of Help Desk in Process Designer Content Pack 9.30.3 and in SM 9.4x Codeless/Hybrid. You need to pay attention to the value mapping about the **Status** field between Service Manager and ALM.

Constants -> SM Fields

In order to create a Problem in SM, you need to specify constant values in field mapping. The constant values vary with different SM versions and SM customizations.

If the Service Manager Process Designer (PD) Content Pack is not installed, or you have installed PD Content Pack 9.30.2 on SM 9.3x, see the following summary for sample constant field mappings:

Note: The following sample mappings also apply to SM 9.4x Classic.

QCS SM constant value	Dir	QCS Name/SM WSDL Caption	SM WSDL Type	SM7 DB Type	SC6 DB Type	SM DB Name / SM WSDL Field	SM Len
1	->	QCIntegrationType	StringType	Char	Text	qcintegration.type	60
Created from Quality Center	->	CreatedFrom	StringType	Char	Text	qcintegration.created.from	60
(your setup)	->	QCProject	StringType	Char	Text	qcintegration.project	60
	->	AssignmentGroup	StringType	Date/time	Date/time	assignment	50
AUTO	->	AssignmentGroup	StringType	Char	Text	assignment	50
client system	->	Category (1)	StringType	Char	Text	incident.category	40
BOB.HELPDESK	->	ProblemOwner	StringType	Char	Text	ticket.owner	40
exchange	->	ProblemType (1)	StringType	Char	Text	problem.type	40
email client	->	ProductType (1)	StringType	Char	Text	product.type	40

Constant -> SM Problem Mappings

Constant -> SM Problem Mappings, continued

QCS SM constant value	Dir	QCS Name/SM WSDL Caption	SM WSDL Type	SM7 DB Type	SC6 DB Type	SM DB Name / SM WSDL Field	SM Len
4 - User	->	Impact	StringType	Char	Text	initial.impact	50
software	->	SubCategory (1)	StringType	Char	Text	subcategory	40
ITIL	->	WorkFlowType	StringType	Char	Text	category	40
Problem Identification and Classification	->	CurrentPhase	StringType	Char	Text	current.phase	40

If the Service Manager Process Designer (PD) Content Pack 9.30.3 is installed on SM 9.3x, see the Constant -> SM Problem Mappings table plus the following row for sample constant field mappings:

QCS SM constant value	Dir	QCS Name/SM WSDL Caption	SM WSDL Type	SM DB type	SM DB Name / SM WSDL Field	SM Len
true	->	IsKnownError	BooleanType	char	isKnownError	1

If you are working with Service Manager 9.40 Codeless or 9.41 Codeless/Hybrid, see the Constant -> SM Problem Mappings table plus the following row for sample constant field mappings:

QCS SM constant value	Dir	QCS Name/SM WSDL Caption	SM WSDL Type	SM DB type	SM DB Name / SM WSDL Field	SM Len
known error	->	Category	StringType	char	Category	40

To create a problem in Service Manager 9.40 Codeless or 9.41 Codeless/Hybrid, map the Category caption in the above table to a problem category.

Define Events

The following table lists the event settings for the two endpoints:

Events Tab Settings	QC Action (Event)	SM Action (Event)
Creation	Create a corresponding record in the other endpoint.	Do nothing.
Update	Update its corresponding record in the other endpoint.	Update its corresponding record in the other endpoint.
Deletion	Do nothing.	Do nothing.

The following screenshot displays the settings:

General Connectivity Scheduling Filters Events Field Mapping Advanced

d in this endpoint
ling record in the other endpoint
ed in this endpoint
nding record in the other endpoint
zation Only)
d from this endpoint
ding record in the other endpoint
its corresponding record in the other endpoint

Test the Link

Follow these steps to test the link:

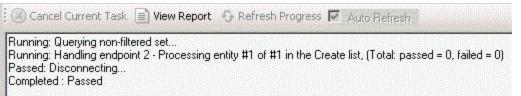
Note:

- 1. 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.
- 2. The following sample steps are for your reference only. The exact steps required on your system may differ significantly. The phase in which the QC Integration tab appears may be different on your system.
- 1. Save the configuration (an integrity check is automatically run).
- 2. Click Enable Link.
- 3. Create a defect and set Synchronize with SM Problem to Y.

🔍 New Def	fect	. 🗆 🗙
× Clear	Attach: 🥢 🥜 📸 🚺 🚔 🕵 🕶 💱 💷 📦	0
* Summary:	defect to problem	
Details	SM Integration (New)	
Sychorniz	ze with SM Problem: Y	
	Submit Close	

Note: In ALM 11, the **Synchronize with SM Problem** field and the **Problem ID** field reside on the Details tab of the Defect form.

4. Synchronize.



5. View the Problem in SM.

lecord Number:		.×M0020		Stat	us:		Open	•
				Expe	ected Resolutio	n Date:		•
rief Description		defect to p	problem					
Classification	Activities	Attachments	♦ Related Records	🔶 History	🔶 Workflow	🔷 QC Integrati	on	
Defe	ct ID:		24					
Serv	er/Domain/Proje	ect	local	host/DEFAULT	/Demo	~		
C	ted from:		Crea	ted from Qual	ity Center			

6. View the Defect in QC.

📴 Defect Deta	nils 📃	
	> 🛠 🏲 🚦 🗉 🗸 💼	0
Defect:	24 defect to problem	
<u>s</u>	Details SM Integration (Details) Sychornize with SM Problem: Y Problem ID: PM0020	_
Details	Sychornize with SM Problem: M Problem ID: PM0020	
- Ol		
Attachments		
æ		
Linked Entities		
\$		
History		
Execution Report	OK Cancel	

Note: In ALM 11, the **Synchronize with SM Problem** field and the **Problem ID** field reside on the Details tab of the Defect form.

Chapter 10: SM Problem <-> QC/ALM Defect

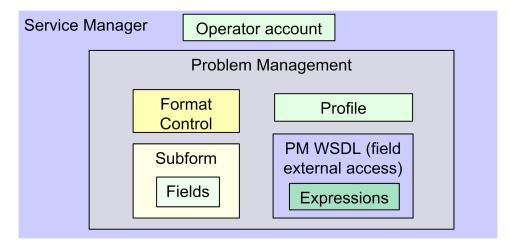
This chapter describes how to synchronize SM Problems with QC/ALM Defects ("Problem <-> Defect").

This chapter includes:

- "Customizing Service Manager for Problem Management " below
- "Customizing the QC/ALM Defects Module" on page 168
- "Configuring Links in QC/ALM Synchronizer" on page 177

Customizing Service Manager for Problem Management

The following diagram summarizes the components which require tailoring in Service Manager.



To customize Service Manager for Problem Management, perform the following tasks:

- 1. "Add Fields" on the next page
- 2. "Specify the External Access Definition on Service Manager" on the next page
- 3. "Create a Subform" on page 156

- 4. "Add the Subform to a Form" on page 158
- 5. "Add Format Control Calculations/Validations" on page 160

Add Fields

Add the following required fields to the rootcause table. Do not change them.

	Туре	Туре				
Field	Service Manager 7.0x or later	ServiceCenter				
qcintegration.type	Character	Text				
qcintegration.id	Number	Decimal				
qcintegration.project	Character	Text				
qcintegration.created.from	Character	Text				

Note: The data type requirements for SM fields are described in "Matching Types" on page 31.

Specify the External Access Definition on Service Manager

If the Service Manager Process Designer (PD) Content Pack is not installed, or you have PD Content Pack 9.30.2 installed, follow the steps below to specify the External Access Definition on Service Manager:

Note: The following steps also apply to SM 9.4x Classic.

- Create a custom External Access Definition QCIntProblemService by clicking Tailoring > WSDL configuration on Service Manager 7.0x; or clicking Tailoring > Web Services > Web Service Configuration on Service Manager 7.1x or later with the following values:
 - Service Name: QCIntProblemService
 - Name: rootcause
 - Object Name: QCIntProblem

Installation and Administration Guide Chapter 10: SM Problem <-> QC/ALM Defect

- Allowed Actions / Action Names:
 - add / Create
 - save / Update

External Access Definition					
Service Name:	QCIntProl	blemService			
Name:	rootcause	:	👻 Obj	ect Name:	QCIntProblem
Allowed Act	ions 🔶	Expressions	🔶 Fields		
Allowed Actio	ns	Action Name	es	Action Typ	ре
add		Create			
save		Update			
add	ns	Create	es	Action Typ	pe

Note: The above values are required (Do NOT change them).

2. Enable the required fields in the web service.

Field	Caption	Туре
id	ProblemID	StringType
sysmodtime	Modified	DateTimeType
qcintegration.id	QCEntityID	IntType
qcintegration.project	QCProject	StringType
qcintegration.type	QCIntegrationType	StringType
qcintegration.created.from	CreatedFrom	StringType
current.phase	CurrentPhase	StringType
category	WorkFlowType	StringType
affected.itemsm.device.display.name	ServiceName	
logical.namesm.device.display.name	AffectedCIName	

External Access De	efinition					
Service Name: QCIntProblemService						
Name: rootca	lse	—				
Allowed Actions	Expressions	Fields				
Field	Caption	Туре				
qcintegration.id	QCEntityID	IntType				
id	ProblemID	StringType				
sysmodtime	Modified	DateTimeType				
qcintegration.project	QCProject	StringType				
incident.category	Category	StringType				
subcategory	SubCategory	StringType				
product.type	ProductType	StringType				
problem.type	ProblemType	StringType				
initial.impact	Impact	StringType				
severity	Severity	StringType				
description	Description	StringType				
assignment	AssignmentGroup	StringType				
ticket.owner	ProblemOwner	StringType				
category	WorkFlowType	StringType				

Note: The affected.item..sm.device.display.name field and the logical.name..sm.device.display.name are available on Service Manager 9.41 only. These fields are used for synchronizing SM Problem to QC/ALM Defects only.

Note: The caption value must be unique and alphanumerics (no spaces) with the first letter capitalized (AValidCaption123, AnotherValidCaption, and so on). The above values are required (Do NOT change them).

3. Define expressions for the web service .

Note:

- a. Problem Management requires an activity update provided with each save and for better flow, and this activity update will be hard-coded with the following expressions.
- b. Expressions 1 to 4 are for fixing an update issue with the Problem Management web service. For more information, see *SCR 41399*.

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	<pre>\$rc.update=update in \$L.file;if (denull(\$rc.update)={}) then (\$rc.update={"QC update sent"})</pre>
3	if (\$L.need.to.update=true) then (\$rc.update={"QC update sent"})
4	update in \$L.file=update in \$L.file.save

External Access Definition

Service Name:	QCIntProblemService			
Name:	rootcause	-	Object Name:	QCIntProblem
♦ Allowed Actions	Fields			
Expressions				
cleanup(\$pm.activity);cleanup(\$r	c.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 (d	denull(\$rc.update)={}) then (\$rc.update={"QC update s	ent"})		
if (\$L.need.to.update=true) ther	n (\$rc.update={"QC update sent"})			
update in \$L.file=update in \$L.file	e.save			
current.phase in \$L.file="Problem	n Investigation and Diagnosis"			
category in \$L.file="ITIL"				

If the Service Manager Process Designer (PD) Content Pack 9.30.3 is installed, follow the steps below to specify the External Access Definition on Service Manager:

Note: The following steps also apply to SM 9.4x Codeless/Hybrid.

- 1. Refer to step 1.
- 2. Enable the required fields in the web service.

Field	Caption	Туре
id	ProblemID	StringType
sysmodtime	Modified	DateTimeType
qcintegration.id	QCEntityID	IntType
affected.itemsm.device.display.name	ServiceName	
logical.namesm.device.display.name	AffectedCIName	

External Access Definition

Service Name: Name:		* QCIntProblemService * rootcause		~	Released:Deprecated:		
Object Name:			QCIntProblem				
	Expressions Fields						
Field	8	Capt	ion	Турс			
id		Problem	nID	StringType			
sysmo	odtime	Modifie	d	DateTimeType			
qcinte	gration.id	QCEntit	tyID	IntType			
qcinte	gration.project	QCProj	ect	StringType			
qcinte	gration.type	QCInteg	grationType	StringType	-		
qcinte	gration.created.from	Created	dFrom	StringType			

Note: The affected.item..sm.device.display.name field and the logical.name..sm.device.display.name are available on Service Manager 9.41 only. These fields are used for synchronizing SM Problem to QC/ALM Defects only.

Note: The caption value must be unique and alphanumerics (no spaces) with the first letter capitalized (AValidCaption123, AnotherValidCaption, and so on). The above values are required (Do NOT change them).

3. Define expressions for the web service.

Note: Problem Management requires an activity update provided with each save and for better flow, and this activity update will be hard-coded with the following expressions.

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	<pre>\$rc.update=update in \$L.file;if (denull(\$rc.update)={}) then (\$rc.update={"QC update sent"})</pre>
3	if (\$L.need.to.update=true) then (\$rc.update={"QC update sent"})
4	update in \$L.file=update in \$L.file.save
5	isKnownError in \$L.file=nullsub(isKnownError in \$L.file,true)

🔜 🗃 OK 💢 Cancel \land Previous 🕅	Next 🜵 Add 💾 Save 🐨 Delete 🔍 Find 🗊 Fill						
External Access Definition record updated.							
External Access Definition							
Service Name:	QCIntProblemService Released						
Name:	rootcause 📼 🗖 Deprecated						
Object Name:	QCIntProblem						
♦ Allowed Actions ♦ Expressions	Fields						
Src.update=update in SL.file; if (SL.need.to.update=true) tl update in SL.file=update in S	p(\$rc.update);if same(update in \$L.file, update in \$L.file.save) then (\$L.need.to.update=true) f (denull(\$rc.update)={}} then (\$rc.update={"QC update sent"}) nen (\$rc.update={"QC update sent"}) L.file.save ub(isKnownError in \$L.file,true)						

Create a Subform

To create a subform:

- 1. Create a global list.
 - a. Click Tailoring > Tailoring Tools on Service Manager7.0x or later; or click Utilities > Tools > Global Lists on ServiceCenter to creaa a global list with the following parameters:

No.	Parameter	Value	Remarks
1	List Name	SMQC Integration PM Project List	
2	Regen Every	1 00:00:00	
3	Build List on Startup?	Yes	Check box
4	List Variable	\$G.qcintegration.problem.project	
5	User Defined List?	Yes	Check box

No.	Parameter	Value	Remarks
6	Value List	{"server1/domain1/project1", "server2/domain2/project2"}	Change to the values for your system
			Note: Do not include any spaces between slashes.

- b. Save this global list and click **Rebuild Global List** from the Options menu.
- Click Tailoring > Forms Designer on Service Manager 7.0x or later; or click Toolkit > Forms
 Designer on ServiceCenter to create the pm.qcint.subform subform with the following components:

Note: Click **No** when the system message "Do you want to use Form Wizard?" appears.

No.	Component	Properties
1	Label	Caption: Synchronize with QC
2	Combo Box	Input: qcintegration.type
		Value List: 0;1;
		Display List: 0 - Not Synchronize;1 - Synchronize with QC Defect
		Select Only: Yes
		Read-Only Condition: [\$qcint.type.readonly]
3	Label	Caption: Defect ID:
4	Text	Input: qcintegration.id
		Read-Only: Yes
5	Label	Caption: Server/Domain/Project:

No.	Component	Properties
6	Combo Box	Input: qcintegration.project
		Value List: \$G.qcintegration.problem.project
		Read-Only Condition: [\$qcint.project.readonly]
		Mandatory Condition: [qcintegration.type]>0
7	Label	Caption: Created from:
8	Text	Input: qcintegration.project
		Read-Only: Yes

🗸 OK 🛛 🗱 Cancel	
	10 🔏 🕂 🛄 📪 🖬 🗔 🜌
Synchronize with QC:	~
Defect ID:	
perect ID:	
Server/Domain/Project	
Created from:	

Add the Subform to a Form

If the Service Manager Process Designer (PD) Content Pack is not installed, or you have PD Content Pack 9.30.2 installed, follow the steps below to add the subform you created to a form:

Note: The following steps also apply to SM 9.4x Classic.

- From Forms Designer, open the default form of a Problem Management phase (PM.pc.ident.and.class is used as an example in ServiceCenter 6.2/Service Manager 7.0x).
- 2. Add a notebook Tab with the QC Integration caption.

3. Add a subform to the new tab with format pm.qcint.subform, which is displayed in the following screenshot:

🚭 *Forms Designer: 🗙 👺 Problem Managemen 🛛 👺 rootcausephase: 🏻 🔭 🗖	Properties 🛛	- 8
	Subfo	ormat
	Property	Value
III III 🖪 🚥 📾 🔣 🗰 🧮 🖉 🏷 🌄 📆 🚼 👗 🚺	Display Blank	
	Display Using Table	
Problem Control - Problem Investigation and Diagnosis	Format	pm.qcint.subform
Record Number:	Height	40
	Input	
Expected Resolution E	Name	
Brief Description:	Virtual Join	
	Visible	
♦ SLA ♦ Related Records ♦ Tasks ♦ History ♦ Workflow ♦ QC Integration »s	Visible Condition	
	Width	143
	X	3
	Y	1

4. Save the changes.

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

If the Service Manager Process Designer (PD) Content Pack 9.30.3 is installed, follow the steps below to add the subform you created to a form:

Note: The following steps also apply to SM 9.4x Codeless/Hybrid.

- 1. From Forms Designer, open the default form of a Problem Management phase (pbm.problem.logging is used as an example on Service Manager 9.3x).
- 2. Add a notebook Tab with the QC Integration caption.
- 3. Add a subform to the new tab with format pbm.problem.logging, which is displayed in the following screenshot:

👺 To Do Queue: My To Do List 🛛 📓 Forms Designer: pm.qcint.subfo 🛛 📓 Forms Designer: pbm.problem.l 🛛 🖓 🗖	Properties 🛛	
👩 OK 💢 Cancel 🔂 🔽 💌		Subformat
	Property	Value
	Name	
	X	5
	Y	2
	Width	72
	Height	22
	Visible	\checkmark
	Visible Condition	
♦ Affected Configur ♦ SLA ♦ Companies ♦ Change Calendar ♦ Attachments ♦ QC Integration »1	Tab Stop	0
	Format	pm.qcint.subform
	Virtual Join	
	Display Blank	\checkmark
	Display Using Table	
	Input	

4. Save the changes.

Add Format Control Calculations/Validations

Note: Steps in this topic are applicable if your Service Manager has no Process Designer (PD) Content Pack installed, or if you are working with SM 9.4x Classic. Otherwise, skip this topic and refer to "Add Rule Set Calculations/Validations" on page 162 for SM 9.3x configurations in the PD environment and SM 9.4x Codeless/Hybrid configurations.

To add format control calculations and validations:

- 1. Open the format control record of the previous Problem Management form PM.pc.ident.and.class.
- 2. Click **Calculations**.
- 3. Add two rows with the following values:

Row	Parameter	Value
1	display	true
	initial	true
	calculation	<pre>\$qcint.type.readonly=2;if (qcintegration.type in \$file~=0) then (\$qcint.type.readonly=1)</pre>
2	display	true
	initial	true
	calculation	<pre>\$qcint.project.readonly=2;if (qcintegration.type in \$file~=0 and not null (qcintegration.project in \$file)) then (\$qcint.project.readonly=1)</pre>

4. Click Validations.

5. Add one validation with the following values:

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

- 6. Save your changes.
- 7. Verify.

Classification	Activities	Attachments	Related Records	🔷 History	🔷 Workflow	🔷 QC Integration	255555
Syr	nchronize with (QC:		1 - Synch	ronize with QC I	Defect	-
Del	ect ID:			7			
Ser	ver/Domain/Pro	oject		localhost/	QADEMO/Demo)	-
Cre	ated from:			Created f	rom Quality Cer	nter	

Add Rule Set Calculations/Validations

Note: Steps in this topic are applicable for the Service Manager 9.3x with Process Designer (PD) Content Pack 9.30.3 and SM 9.4x Codeless/Hybrid only. Otherwise, refer to "Add Format Control Calculations/Validations" on page 160 for configurations in SM 9.3x non-PD, SM 9.3x with PD Content Pack 9.30.2, and SM 9.4x Classic environment.

To add rule set calculations and validations, follow the steps below:

- 1. "Copy an Existing Workflow" below
- 2. "Associate an Existing Problem Category with the New Workflow" on the next page
- 3. "Create New Rule Set for Initialization and Validation" on page 164
- 4. "Associate the New Workflow with the New Rule Set" on page 166

Copy an Existing Workflow

You can use copies of the existing workflows in another business process, or make changes to the HP proprietary workflow copies.

To copy an existing workflow, follow the steps below:

- 1. From the System Navigator, click **Tailoring > Process Designer > Copy Existing Workflow**.
- 2. On the Clone a Workflow page, select the workflow you want to copy. For example, Problem.

3. Type SMQCIntPbM in the New workflow name field.

Clone	a Workflow		
		Please specify the new workflow name, as well as the p	refix for new rule sets if they are to be copied as well.
		New workflow name:	* SMQCIntPbM
		Copy rule sets?	V
		Rule set prefix:	* SMQCIntPbM

- 4. Select the Copy rule sets check box if you want to copy rule sets, and then type a rule set prefix.
- 5. Click **OK**.

The newly copied workflow appears in the list on the Clone a Workflow page.

Associate an Existing Problem Category with the New Workflow

You can update existing problem categories, subcategories and areas and associate the updated categories with the new workflow so that they can be used in another business process. For more information, refer to *HP Service Manager – Process Designer Content Pack Administrator's Guide*.

To associate an existing change category with the new workflow, follow the steps below:

- 1. From the System Navigator, click **Problem Management > Configuration > Problem Categories**.
- 2. Click Search.
- 3. Select the problem category for which you want to add a workflow. For example, problem.
- 4. In the Problem Category page, remove the currently assigned workflow from the Workflow field.
- 5. Type SMQCIntPbM in the workflow field.

6. Click **Save** to associate the problem category with the workflow.

Problem Category				
Name:	problem	Apply To:	Problem	
Active:	V			
Description:	incident			
Workflow:	* SMQCIntPbM	đQ		
Subcategories Workflow				٩
Logging	Categorization	¢ Resolution ¢ Review Review	Closure	

Create New Rule Set for Initialization and Validation

To create a new rule set for initialization, follow the steps below:

- 1. From the System Navigator, click **Tailoring > Process Designer > Rule Sets**.
- 2. Type the values as follow:

Field	Value
ID	pbm.alm.int.init
Available as action	False
Name	Initialize for ALM integration in the Problem Record
Table name	rootcause
HP Proprietary	

- 3. Click New and Save.
- 4. Click Add Rule.
- 5. In the Select Rule Type page, click Run JavaScript.

Installation and Administration Guide Chapter 10: SM Problem <-> QC/ALM Defect

6. In the Run JavaScript page, type the values as follow:

Field	Value
Rule Description	Run Javascript for initializing Integration type and project in the Problem Record
Statement	<pre>vars['\$qcint.type.readonly'] = 2; vars['\$qcint.project.readonly'] = 2; var _null=system.functionsnull; var file = vars.\$L_file; if(file["qcintegration.type"] !=0 && !_null(file["qcintegration.type"])) { vars['\$qcint.type.readonly'] = 1 } if(file["qcintegration.type"] !=0 && !_null(file["qcintegration.project"])) { vars['\$qcint.project.readonly'] = 1 }</pre>

7. Click **OK**.

8. Click **Save** and **Exit**.

To create a new rule set for validation, follow the steps below:

- 1. From the System Navigator, click **Tailoring > Process Designer > Rule Sets**.
- 2. Type the values as follow:

Field	Value
ID	pbm.alm.int.validation
Available as action	False
Name	Validation for ALM integration in the Problem Record

Field	Value
Table name	rootcause
HP Proprietary	

- 3. Click New and Save.
- 4. Click Add Rule.
- 5. In the Select Rule Type page, click **Set Mandatory Fields**.
- 6. Refer to step 6 to step 13 as described in "Create New Rule Set for Initialization and Validation" on page 60.

Associate the New Workflow with the New Rule Set

To associate the new workflow with the new initialization rule set, follow the steps below:

- 1. From the System Navigator, click **Problem Management > Configuration > Workflows**.
- 2. Select Problem in the workflows list.
- 3. Select the first phase in the workflow graph.
- 4. Click **Rule Sets** tab > **Initialization** tab.
- 5. Click **Add** and select the pbm.alm.int.init rule set you just created.

HP Service Manager									User: falcon
		To Do Queue: My To Do List Workflows	Wo	kflow: Problem 🛎 Rule Sets					
	«	💾 Save 🏵 Zoom in 🤤 Zoom out 🔲 Ad	ld phase	🐨 Delete 🛛 🚰 Workflow properties					8 🖬 🔤
Open New Problem Problem Queue									
Problem Task Queue									
Search Knowledgebase Search Problems Search Problem Tasks					<u> </u>				
Request Management		Logging Categorizati	ion	C Investigation	Resolution	Review Closure			
Service Catalog									
Service Desk									
Service Level Management		Abandon	me-t	Rule Sets - Initialization					
System Administration				Id		Name			
Tailoring				apply.template		Apply Template			
> Audit				apply.template create.template		Create Template from Record			
Differential Upgrade		•						1	•
Document Engine		Phase - Logging		pbm.abandon.wizard		Run Abandon Wizard			
Event Services				pbm.abandonment		PBM abandonment			
Knowledge Engineering		Details Forms Rule Sets Actions		pbm.abandonment.field.validation		PBM field validation at abandonment phase			
Notifications	=	On enter On exit Initialization		pbm.alm.int.init		Initialize for ALM integration in the Problem Record			
Process Designer			10	pbm.bac.pi.assciate		Associate with BAC PI			
Configuration		Add 💮 Delete 🙃 View 🐴 Up 😺 Do	W	pbm.calculate.rc.calendar		Calculate whether RC calendar needs to be displayed			
Copy Existing Workflow Export Workflow		Rule Sets		pbm.categorization.init.status		Initialize pm status at categorization phase			
Rule Sets		pbm.status.list		pbm.categorization.mandatory		PBM mandatory validation at categorization phase			
Workflows		pbm.set.impact.urgency.value		ponteriogonization in tradition y		i bin manaatory randanon at oatogonization phaoo			
 SQL Utilities 		pbm.logging.init.status				OK Can	el		
Tailoring Tools		pbm.initialize.id							
> Web Services		E pontinuario o							
Codes									
Database Dictionary									

- 6. Click **OK**.
- 7. Click Rule Sets tab > On display tab.
- 8. Repeat step 5 and 6.
- 9. Click **Rule Sets** tab > **On enter** tab.
- 10. Click Add and select the pbm.alm.int.validation rule set you just created.

MP Service Manage	er				Us
		To Do Queue: My To Do List Workflows	Workflow: Problem 🗵		
🗟 🔯 🚭	*	💾 Save 🔍 Zoom in 🍳 Zoom out 🔲 Add	phase 💮 Delete 🚰 Workflow properties		8
Open New Problem Problem Queue Problem Task Queue Search Knowledgebase Search Problem Tasks Request Management Service Catalog	*			slution Review - Cloure	
Service Desk					
Service Level Management		Abandonm			
system Administration			Add Rule Sets - On enter		
alloring			ld Id	Name	
Audit			apply.template	Apply Template	<u> </u>
Differential Upgrade		•	create.template	Create Template from Record	=
Document Engine		Phase - Logging	pbm.abandon.wizard	Run Abandon Wizard	
Event Services			pbm.abandonment	PBM abandonment	
Knowledge Engineering		Details Forms Rule Sets Actions	pbm.abandonment.field.validation	PBM field validation at abandonment phase	
Notifications	=	On enter On exit Initialization O	pbm.alm.int.init	Initialize for ALM integration in the Problem Record	
Process Designer			pbm.alm.int.validation	Validation for ALM integration in the Problem Record	
Configuration		🖨 Add 🗑 Delete 🙃 View 🏫 Up 😺 Dov	pbm.bac.pi.assciate	Associate with BAC PI	
Copy Existing Workflow		Rule Sets	pbm.calculate.rc.calendar	Calculate whether RC calendar needs to be displayed	
Export Workflow		pbm.logging.mandatory	pbm.categorization.init.status	Initialize pm status at categorization phase	
Rule Sets Workflows		bbm.set.openedby	poincategorization.thit.status	initianze prinstatus at categorization priase	T
SQL Utilities		pbm.set.incidentcategory		OK Can	cel
Tailoring Tools		pbm.set.affected.cis			
Web Services					
Codes		pbm.init.sla			
Database Dictionary					
Database Manager					

- 11. Click **OK**.
- 12. Click Save.

Customizing the QC/ALM Defects Module

The steps for customizing the Defects module are different for different versions of Quality Center/ALM.

On QC 10 and Earlier

To customize the Defects module on Quality Center 10 or earlier, perform the following tasks:

- 1. "Add Fields" below
- 2. "Add Tabs" on the next page
- 3. "Add Fields to Tabs" on page 171
- 4. "Create a View" on page 173
- 5. "Verify" on page 173

Add Fields

To add the required fields for Defect module customization:

- 1. Log on to QC as a project administrator.
- 2. Click **Tools / Customize**. The "QC Project Customization" module opens.
- 3. Add the following fields for the defect entity in Project Entities (*XX*, *XY* and *XZ* are sequential numbers auto-generated by QC).

Field Name	Field Label	Field Type	Remarks
BG_USER_ <i>XX</i>	Synchronize with SM Problem	Lookup List/YesNo	Select "Verify Value" check box
BG_USER_ <i>XY</i>	Problem ID	String	
BG_USER_ <i>XZ</i>	Created from	String	

Quality Center -	Project Customization			
<u>User Properties</u> Project Users <u>Groups</u> Machila Researc	Project Entities	Field Settings		
Module Access Project Entities Requirement Types Risk-Based Quality Project Lists Automail Alert Rules Workflow	Defect Defect System Fields User Fields Change id Forward as problem Rec Problem ID	Field Name: Field Label: Field Type: Field Length:	BG_USER_01 Forward as problem Lookup List 40 Required	·
		Masked	Searchable	

Note: The data type requirements for QC fields is described in "Matching Types" on page 31.

Add Tabs

To add tabs to the Defect form and show fields on these tabs:

- 1. In "QC Project Customization", click **Workflow** > **Script Editor**.
- 2. Select **Defects module script**.

User Properties Project Users Workf	low
Module Access Enables Project Entities You can Requirement Types Script Ge Risk-Based Quality Man Enables	enerator - Add Defect Field Customization you to customize the fields displayed for each user group in the Add Defects dialog bo also specify field order and whether a field is required. enerator - Defect Details Field Customization you to customize the fields displayed for each user group in the Defect Details dialog can also specify field order and whether a field is required.
Alert Rules Script Ed Workflow Enables You can	litor you to write VBScript code for all Quality Center modules. also use the Script Editor to modify the scripts generated by the above tools.
Script Editor Script Editor Toolbar Button Editor	GetNewBugPageName
- S Workflow Scripts	GetDetailsPageName
😥 📄 Common script	
Requirements module script	
Test Plan module script	
⊕ B Test Lab module script ⊕ B Manual Runner script	
Defects module script	Sug_canterPost →Sug_AfterPost

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

Note: The parameter 2 specifies tab 2 (the second tab). For a new bug, the tab name is SM Integration (New).

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

select case PageNum

case "2"

GetDetailsPageName = "SM Integration (Details)"

end select

Note: The parameter 2 specifies tab 2 (the second tab). For an existing defect, the tab name is SM Integration (Details).

Add Fields to Tabs

To add fields to tabs:

- 1. In "QC Project Customization", click **Workflow** > **Script Editor**.
- 2. Select **Defects module script**.

User Properties Project Users Workflo			w
Groups Script Gen Module Access Enables y Project Entities You can al Requirement Types Script Gen Risk-Based Quality Man Enables y Project Lists Enables y		Enables y You can a Script Gen Enables y	nerator - Add Defect Field Customization ou to customize the fields displayed for each user group in the Add Defects dialog bo lso specify field order and whether a field is required. Interator - Defect Details Field Customization ou to customize the fields displayed for each user group in the Defect Details dialog an also specify field order and whether a field is required.
Alert Rules Workflow		,	tor ou to write VBScript code for all Quality Center modules. Iso use the Script Editor to modify the scripts generated by the above tools.
Script Edit	or		so and the completation to moting the complet generated by the debite loois.
Script Edit	or Toolbar But		GetNewBugPageName

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

a. Script Generator - Add Defect Field Customization

b. Script Generator - Defect Details Field Customization



4. 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 consists of two digits)
- Visible (True)
- Required (False)
- Page number (start from 0)
- View order (start from 0)
- 5. 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

6. Set the **Readonly** fields by adding the following lines to the **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

The if loop above marks the field "Synchronize with SM Problem" as read-only after selected and saved.

7. Save your changes.

Create a View

To create a view:

- 1. Log on to ALM with the integration account SMQCIntUser.
- 2. In the Defects module, click **View / Filter/Sort / Set Filters/Sort**. The purpose of this view is to make the QC Synchronizer correctly filter those defects to be synchronized to SM as problems.
- 3. Set Synchronize with SM Problem to Y.
- 4. Add a view to Favorites:
 - Name: SMIntegrationView
 - Location: Private

Add Favorite						
Add Favorit	e					
Name:	SMIntegration∀iew					
Location:	 Private 	0	Public			

In QC Synchronizer this view will be selected as the QC data filter. Without this filter, QC defects cannot be forwarded to SM as Problems.

Verify

Refer to the following screenshot to verify whether the Defects module on Quality Center 10 or earlier is customized successfully:

Details SM Integration (Details)			
Synchronize with SM Problem: 👔		Problem ID:	PM0017
Created from: Cr	reated from SM/SC		

On ALM 11

On ALM 11, you only need to add new fields directly to the Details tab of the Defect form.

To customize the ALM Defects module, perform the following task:

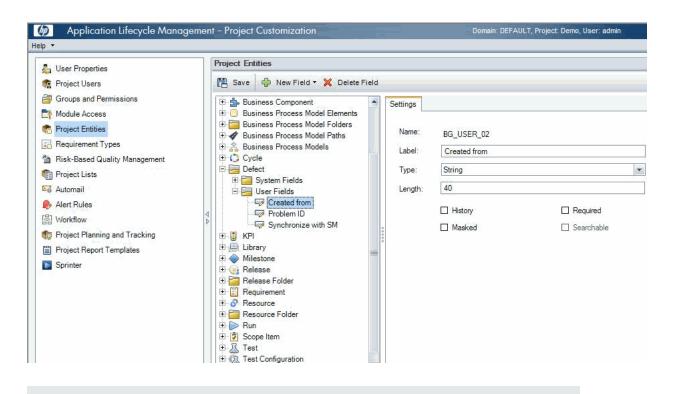
- 1. "Add Fields" below
- 2. "Create a View" on page 176
- 3. "Verify" on page 176

Add Fields

To add the required fields for Defect customization:

- 1. Log on to ALM as a project administrator.
- Click Tools / Customize. The "Application Lifecycle Management Project Customization" module opens.
- 3. Add the following fields for the defect entity in Project Entities (*XX* and *XY* are sequential numbers auto-generated by ALM).

Field Name	Field Label	Field Type	Remarks
BG_USER_ <i>XX</i>	Synchronize with SM Problem	Lookup List/YesNo	Select the "Verify Value" check box
BG_USER_ <i>XY</i>	Problem ID	String	



Note: The data type requirements for QC/ALM fields are described in "Matching Types" on page 31.

- 4. Click Workflow > Script Editor.
- 5. Select Defects module script.
- Set the **Readonly** fields by adding the following lines to the **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

The if loop above marks the field "Synchronize with SM Problem" as read-only after selected and saved.

7. Save your changes.

Create a View

To create a view:

- 1. Log on to ALM with the integration account SMQCIntUser.
- 2. In the Defects module, click **View / Filter/Sort / Set Filters/Sort**. The purpose of this view is to make the ALM Synchronizer correctly filter those defects to be synchronized to SM as problems.
- 3. Set Synchronize with SM Problem to Y.
- 4. Add a view to Favorites:
 - Name: SMIntegrationView
 - Location: Private

Add Favoril	te	×
Add Favorit	te	
Name:	SMIntegration∀iew	
Location:	Private O Public	:

In ALM Synchronizer this view will be selected as the ALM data filter. Without this filter, ALM defects cannot be forwarded to SM as Problems.

Verify

Open a new defect in ALM, select **Y** in the **Synchronize with SM** field, and click **Save**. If the Defects module on ALM 11 is customized successfully, the Defect form is displayed as follow:

Details					
Actual Fix Time:			Synchronize with SM:	Y	
Closing Date:		•	Closed in Version:		•
* Detected By:	SMQCIntUser	₩ .	Detected in Cycle:		•
Detected in Release:		•	* Detected on Date:	2010-11-12	•
Detected in Version:		•	Estimated Fix Time:		
Planned Closing Ver		•	Priority:		•
Project:		•	Reproducible:	Y	•
Assigned To:		•	* Severity:	2-Medium	•
Status:	New	•	Subject:		•
Target Cycle:		•	Target Release:		•
Change ID:			Created from:		
Problem ID:			Modified:		

Configuring Links in QC/ALM Synchronizer

To configure and test a link in the QC/ALM synchronizer, perform the following tasks:

- "Specify Endpoints / Type of Link" below
- "Define Filters" on the next page
- "Define Field Mappings" on the next page
- "Define Events" on page 181
- "Test the Link" on page 182

Specify Endpoints / Type of Link

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

- 1. Step 1: "Endpoint 2 type" = SM ProblemManagement.
- 2. Step 3: "Service URL" =

http://<*service_manager_host*>:<*port*>/sc62server/PWS/QCIntProblemService.wsdl

3. Step 4: "Select entity types" = Problem by Defect (this is the only available selection)

Define Filters

On the Filters tab, select filter **SMIntegrationView** for the QC endpoint. If the filter is not available, see "Create a View" on page 135.

_ QC	SM ProblemManagement
C No Filter	 No Filter
 Use filter (for creation events): 	C Use filter (for creation events):
Private: SMIntegrationView	

Define Field Mappings

If the Service Manager Process Designer (PD) Content Pack is not installed, or you have installed PD Content Pack 9.30.2, see the following summary for basic field mappings:

Note: The following mappings also apply to SM 9.4x Classic.

QC	Directions	SM	Constant Value	Remarks
Problem ID	<-	ProblemID		Synchronize back on create: Yes
Defect ID	->	QCEntityID		Synchronize back on create: Yes
Synchronize with SM Problem			Y	
		QCIntegrationType	1	
Created from			Created from SM/SC	
		CreatedFrom	Created from QC	

QC	Directions	SM	Constant Value	Remarks
	->	CurrentPhase	XXX	Replace <i>XXX</i> with a valid phase name, such as "Problem Investigation and Diagnosis".
				This field mapping is optional for demo data of Service Manager 7.10.
	->	QCProject	(your setup)	This value should be same with the "QC Project" parameter in the Connectivity tab.
	->	WorkFlowType	YYY	Replace YYY with a valid category name, such as <i>ITIL</i> for demo data of SM 7.0x/SC 6.2; <i>BPPM</i> is for demo data of Service Manager 7.10. This field mapping is optional for demo
				data of Service Manager 7.10.

Sample field mappings are shown in the following screenshot:

Mapped Fields					
Туре	QC Field	Direction	SM ProblemManagement Field		
6	Summary	<>	Description		
6	Defect ID	>	QCEntityID		
G	Severity	<>	Severity		
G	Problem ID	<	ProblemID		
~	Sychornize with SM Problem	<	Value: Y		
8	Value: Created from Quality C	>	CreatedFrom		
<u> </u>	Value: 1	>	QCIntegrationType		
8	Value: AUTO	>	AssignmentGroup		
~	Value: BOB.HELPDESK	>	ProblemOwner		
8	Value: client system	>	Category		
~	Value: software	>	SubCategory		
<i>\</i>	Value: email client	>	ProductType		
~	Value: outlook	>	ProblemType		
8	Value: 4 - User	>	Impact		
~	Value: Problem Identification	>	CurrentPhase		
<i>\</i> ∂	Value: localhost/DEFAULT/	>	QCProject		
()	Value: ITIL	>	WorkFlowType		
8	Created from	<	Value: Created from SM/SC		

If the Service Manager Process Designer (PD) Content Pack 9.30.3 is installed, see the above table plus the following row for basic field mappings:

QC	Directions	SM	Constant Value	Remarks
	->	lsKnownError	true	This field mapping is added in the PD Content Pack 9.30.3. This field mapping marks the current record as KnownError.

Sample field mappings are shown in the following screenshot:

Mapped Fields					
Туре	QC Field	Direction	SM ProblemManagement Field		
D	Summaiy	<>	Description		
D	Defect ID	>	QCEntityID		
<u> </u>	Severity	<>	Severity		
G	Problem ID	<	ProblemID		
<i>\</i>	Sychomize with SM Problem	<	Value: Y		
8	Value: Created from Quality C.,.	>	CreatedFrom		
- {∂}	Valva: 1	>	QCIntegrationType		
8	Value: AUTO	>	AssignmentGroup		
8	Value: BOB.HELPDESK	>	ProblemOwner		
<i>\</i>	Value: client system	>	Category		
<	Value: toltware	>	SubCategory		
8	Value: email client	>	ProductType		
<i>\</i>	Value: outlook	>	ProblemType		
<	Value: 4 - User	>	Impact		
<i>\</i>	Value: Problem (dentification	>	CurrentPhase		
8	Value: localhost/DEFAULT/	>	QCProject		
<i>\</i>	Value: ITIL	>	WorkFlowType		
8	Created from	<	Value: Created from SM/SC		
<	Value: true	>	IsKnownError		

Define Events

The following table lists the event settings for the two endpoints.

Events Tab Settings	QC Action (Event)	SM Action (Event)
Creation	Create a corresponding record in the other endpoint.	Create a corresponding record in the other endpoint.
Update	Update its corresponding record in the other endpoint.	Update its corresponding record in the other endpoint.
Deletion	Do nothing.	Do nothing.

The following screenshot displays the settings:

Installation and Administration Guide Chapter 10: SM Problem <-> QC/ALM Defect

General Connectivity Scheduling Filters Events Field Mapping Advan	ced
	SM ProblemManagement
Creation	Creation
When a record is created in this endpoint	When a record is created in this endpoint
Create a corresponding record in the other endpoint	C Create a corresponding record in the other endpoint
Do nothing	C Do nothing
Update	Update
When a record is updated in this endpoint	When a record is updated in this endpoint
Update its corresponding record in the other endpoint	Update its corresponding record in the other endpoint
Do nothing	Do nothing
Deletion (Full Synchronization Only)	Deletion (Full Synchronization Only)
When a record is deleted from this endpoint	When a record is deleted from this endpoint
Do nothing	Do nothing
Delete its corresponding record in the other endpoint	Delete its corresponding record in the other endpoint
Recreate based on its corresponding record in the other endpoint	Recreate based on its corresponding record in the other endpoint

Test the Link

To test the link:

Note:

- 1. 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.
- 2. The following sample steps are for your reference only. The exact steps required on your system may differ significantly. The phase in which the QC Integration tab appears may be different on your system.
- 1. Save the configuration (an integrity check is automatically run).
- 2. Click Enable Link.
- 3. Create a problem in SM and select "1-Synchronize with QC Defect".

							2	\square
Problem Cont	rol - Probler	n Identificatic	on and Classificat	ion				
Record Number:		PM0021		Stat		l	Updated	-
Brief Description:		defect <->	> problem new problen		ected Resolutio	n Date:		 ▼
Classification	Activities	Attachments	Related Records	🔶 History	🔷 Workflow	🔷 QC Integratio	on	
Sync	hronize with QC	:	1 - 5	ynchronize wi	th QC Defect	v		
Defe	ct ID:							
Servi	er/Domain/Proje	ct	local	host/DEFAUL1	ſ/Demo	-		
Crea	ted from:							

4. Create a defect in QC and set Synchronize with SM Problem to **Y**.

New Defect	
🗙 Clear Attach: 🥒 🥜 📸 🛐 🚔 🐢 - 🚏 💷 👹	0
* Summary: defect <-> problem new defect	
Details SM Integration (New)	
Sychornize with SM Problem: The Problem ID:	
Created from:	
Submit Close	

5. Synchronize.

Cancel Current Task in View Report Refresh Progress Auto Refresh

Running: Task execution started.

Running: Connecting to endpoint 1...

Running: Querying non-filtered set...

Running: Handling endpoint 2 - Processing entity #1 of #1 in the Create list, (Total: passed = 0, failed = 0)

Running: Handling endpoint 1 - Processing entity #1 of #1 in the Create list, (Total: passed = 2, failed = 0)

Passed: Disconnecting...

Completed : Passed

6. View the problem in SM.

roblem Control - Prob	olem Identification an	d Classification			
Record Number:	PM0021	Status:		Updated	~
		Expected Re	esolution Date:		-
rief Description:	defect <-> probl	em new problem			
Classification	es 🔷 Attachments 🗇 Ri	elated Records 🛛 🗇 History 🔷 Wor	'kflow 🛛 🧇 QC Inted	ration	
			v qe intog		
Synchronize with	OC:	1 - Synchronize with QC De	efect	~	
·				×	
Defect ID:		26			
				_	
Server/Domain/Pr	roject	localhost/DEFAULT/Demo		V	
				-	
Created from:					
Created from:					
Created from:					
Problem Control - Prol	blem Identification a	nd Classification			
	blem Identification an PM0022	Status:		Open	· ·
roblem Control - Prol Record Number:	<u>ک</u> M0022	Status: Expected R	esolution Date:	Open	▼ ▼
Problem Control - Prol Record Number:		Status: Expected R	esolution Date:	Open	
Problem Control - Prol Record Number: Brief Description:	אינטער איז און	Status: Expected R	esolution Date:		
Problem Control – Prol Record Number: Brief Description:	אינטער איז און	Status: Expected R			
Problem Control - Prol Record Number: Brief Description:	אינטער איז און	Status: Expected R			
Problem Control - Prol Record Number: Brief Description:	[>M0022 defect <-> prot	Status: Expected R	orkflow 🛛 🔷 QC Inte		
Problem Control - Prol Record Number: Brief Description: Classification	[>M0022 defect <-> prot	Status: Expected R blem new defect telated Records 🗇 History 🗇 Wo	orkflow 🛛 🔷 QC Inte	gration	
Problem Control - Prol Record Number: Brief Description: Classification 🛛 🕹 Activiti	[>M0022 defect <-> prot	Status: Expected R olem new defect Related Records 🔷 History 🔷 Wo	orkflow 🛛 🔷 QC Inte	gration	
Problem Control - Prol Record Number: Brief Description: Classification	>M0022 defect <-> prot ies ♦ Attachments ♦ F h QC:	Status: Expected R blem new defect telated Records 🗇 History 🗇 Wo	orkflow 🛛 🔷 QC Inte	gration	
Problem Control - Pro Record Number: Brief Description: Classification 🗇 Activiti Synchronize with Defect ID:	>M0022 defect <-> prot ies ♦ Attachments ♦ F h QC:	Status: Expected R olem new defect Related Records 🗇 History 🗇 Wo 1 - Synchronize with QC D	orkflow 🛛 🔷 QC Inte	gration	

7. View the defect in QC.

📑 Defect Det	ails	_ 🗆 🗙
	▶ 🞗 🏲 ! 🗉 • 🗊	0
Defect:	25 defect <-> problem new defect	
2	Details SM Integration (Details) Sychornize with SM Problem: Problem ID: PM0022	
Details	Sychornize with SM Problem: Problem ID: PM0022	-
°Cij	Created from:	
🗒 Defect Deta	ails	
		0
Defect:	26 defect <-> problem new problem	
	Details SM Integration (Details)	
Details	Sychornize with SM Problem: Y Problem ID: PM0021	
17	Created from: Created from SM/SC	
Attachments		
â		
Linked Entities		
\$		
History		

Note: In ALM 11, the **Synchronize with SM Problem** field and the **Problem ID** field reside on the Details tab of the Defect form.

Chapter 11: Upgrading the Integration

This chapter describes how to upgrade the integration. If the integration components such as Service Manager/ServiceCenter, QC/ALM, and QC/ALM Synchronizer need to be upgraded, refer to the Upgrade Guides of these products.

This chapter includes:

- "Upgrading to the Latest Release" below
- "Post-Upgrade Tasks" on page 198

Upgrading to the Latest Release

To upgrade the integration to the latest release, perform the following tasks:

- 1. "Back up Jar Files and Links" below
- 2. "Deploy the Latest Adapters" below
- 3. "Upgrade for Various Synchronization Scenarios" on the next page

Back up Jar Files and Links

To back up jar files and links:

- 1. Back up all sm-*.jar files (including stub jar) in the <*QCS_Install_Dir*>\adapters\lib directory.
- Back up the configurations of all links in the QCS client by clicking Link > Export > Link Configuration Into XML File....
- 3. Back up the data of all links in the QCS client by clicking Link > Export > Link Data Into Backup File....

Deploy the Latest Adapters

To deploy the latest adapters:

- 1. Stop the HP Synchronizer server by clicking **All Programs > HP Quality Center Synchronizer > Stop Synchronizer**.
- 2. Remove all sm-*.jar files from the <*QCS_Install_Dir*>\adapters\lib directory.
- Install the SMQC Patch 2 package.
 Download the installer for Patch 2 from the HP Quality Center Add-ins website (http://updates.merc-int.com/qualitycenter/qc90/sync/sm/index.html) and run it.
- Deploy all jars in the <*release-package*>\adapter folder and the stub jar to the <*QCS_Install_Dir*>\adapters\lib directory. See "Deploying the Adapters" on page 16 for more details.
- 5. Start the HP Synchronizer server by clicking All Programs > HP Quality Center Synchronizer > Start Synchronizer.

Upgrade for Various Synchronization Scenarios

This section describes how to upgrade various synchronization scenarios to the latest release.

This section includes:

- "Upgrade in Non-PD Environment" below
- "Upgrade from Non-PD Environment to PD Content Pack 9.30.2 (for SM 9.3x only)" on page 193
- "Upgrade from Non-PD Environment to PD Content Pack 9.30.3/SM 9.4x Codeless " on page 193
- "Upgrade from 9.3x with PD Content Pack 9.30.2 to 9.30.3/SM 9.4x Codeless/SM 9.4x Hybrid" on page 195

Upgrade in Non-PD Environment

This section describes how to upgrade various synchronization scenarios to the latest release in SM non-PD environment.

This section includes:

- "SM Change -> QC/ALM Defect" on the next page
- "SM Change -> QC/ALM Requirement" on page 189
- "SM Problem -> QC/ALM Defect" on page 190

- "QC/ALM Defect -> SM Problem" on page 190
- "SM Problem <-> QC/ALM Defect" on page 193

SM Change -> QC/ALM Defect

To upgrade the synchronization scenario of "SM Changes to QC/ALM Defects" ("Change -> Defect"), perform the following tasks:

1. Upgrade SM/SC Customization

No upgrade required.

2. Upgrade QCALM Customization

Perform the following steps to add a new field "Created from" in the defect entity:

- a. Log on to QC as a project administrator.
- b. Click **Tools / Customize**. The "QC Project Customization" module opens.
- c. Add the following field for the defect entity in Project Entities (*XY* is a sequential number autogenerated by QC).

Field Name	Field Label	Field Type	Remarks
BG_USER_XY	Created from	String	

- d. In the "QC Project Customization" module, click Workflow.
 - i. Add the following code to the **WizardFieldCust_Details** and **WizardFieldCust_Add** event procedures

SetFieldApp "BG_USER_XY", True, False, 1, 1

ii. Set the field to Read-only by adding the following lines to the **Bug_New** and **Bug_Moveto** subroutines.

Bug_Fields.Field("BG_USER_XY").IsReadOnly=True

e. Save your changes and log out.

3. Upgrade QC/ALM Synchronizer Customization

Perform the following steps:

- a. In the Quality Center Synchronizer client, edit the link and refresh Schemas.
- b. Add the following constant mapping to the link for this synchronization scenario.

QC	Direction	SM	Constant Value
Created from	<-		Created from SM/SC

c. Save the link.

SM Change -> QC/ALM Requirement

To upgrade the synchronization scenario of "SM Changes to QC/ALM Requirements" ("Change -> Requirement"), perform the following tasks:

1. Upgrade SM/SC Customization

No upgrade required.

2. Upgrade QC/ALM Customization

Perform the following steps to add a new field "Created from" in the requirement entity:

- a. Log on to QC as a project administrator.
- b. Click **Tools / Customize**. The "QC Project Customization" module opens.
- c. Add the following fields for the requirement entity in Project Entities (*XY* are sequential numbers auto-generated by QC).

Field Name	Field Label	Field Type
RQ_USER_XY	Created from	String

- In Requirement Types add the "Created from" field to the Business type requirement.
 The Business type is the default requirement type for incoming requirements (other types can be used).
- e. In the "QC Project Customization" module, click **Workflow**.

In the Script Editor for the Requirements module script, add the following code to **Req_New** and **Req_Moveto** to set fields as read-only and place the fields on the tabs (**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_XY").IsReadOnly=True

SetReqField "RQ_USER_XY", True, False, 1, 1

f. Save your changes and log out.

3. Upgrade QC/ALM Synchronizer Customization

Perform the following steps:

- a. In the Quality Center Synchronizer client, edit the link and refresh Schemas.
- b. Add the following constant mapping to the link for this synchronization scenario.

QC	Direction	SM	Constant Value
Created from	<-		Created from SM/SC

c. Save the link.

SM Problem -> QC/ALM Defect

No upgrade required.

QC/ALM Defect -> SM Problem

To upgrade the synchronization scenario of "QC/ALM Defects to SM Problems" ("Defect -> Problem"), perform the following tasks:

1. Upgrade SM/SC Customization

Perform the following steps:

- a. Log on to SM/SC as a system administrator.
- b. Click System Definition > Tables > rootcause > Fields.
- c. Add two new fields as follow:

	Туре				
Field	Service Manager 7.0x or later	ServiceCenter 6.2			
qcintegration.type	Character	Text			
qcintegration.created.from	Character	Text			

d. Customize forms.

Add a field "Created From" to subform "pm.qcint.subform". See "Add Fields" on page 123.

e. Configure WSDL.

Open the "QCIntProblemService" service in WSDL Configuration, and enable the following two fields:

Field	Caption	Туре
qcintegration.type	QCIntegrationType	StringType
qcintegration.created.from	CreatedFrom	StringType

See "Specify the External Access Definition on Service Manager" on page 123 and Specify the External Access Definition on ServiceCenter for more information.

f. Restart the server if you are using ServiceCenter 6.2.

2. Upgrade QC/ALM Customization

Perform the following steps to add a new field "Created from" in the defect entity:

- a. Log on to QC as a project administrator.
- b. Click **Tools / Customize**. The Module "QC Project Customization" module opens.
- c. Rename "Forward as Problem" to "Synchronize with SM Problem"

Project Entities	- Field Settings	
⊕∽∰ Cycle ⊖∽∰ Defect	Field Name:	BG_USER_01
⊞ <u>e-</u> System Fields ⊟ <u>e-</u> User Fields	Field Label:	Synchronize with SM Problem
·····nec Change ID ·····nec Created from	Field Type:	Lookup List 💌
Froblem ID Synchronize with SM Problem	Field Length:	40
∃∰ Release ∃∰ Release Folder		
E	History	🔲 Required
E- III Run	Masked	🗌 Searchable
⊒∰ Test ⊒∰ Test Instance ⊒∰ Test Set	Lookup List	
	YesNo	👻 New List Goto List

- d. Log off and log on again with the integration account.
- e. Update filter "SMIntegrationView" to "Synchronize with SM Problem = Y".



f. Log off.

3. Upgrade QC/ALM Synchronizer Customization

Perform the following steps:

- a. Deploy the stub jar again. See "Generating/Deploying the Stub" on page 17.
- b. In the QC/ALM Synchronizer client, edit the link and refresh Schemas.
- c. Add the following two constant mappings to the link for this synchronization scenario.

QC	Direction	SM	Constant Value
	->	QCIntegrationType	1
	->	CreatedFrom	Created from Quality Center

d. Save the link.

SM Problem <-> QC/ALM Defect

No upgrade required.

Upgrade from Non-PD Environment to PD Content Pack 9.30.2 (for SM 9.3x only)

This section describes how to upgrade various synchronization scenarios from SM 9.3x non-PD environment to SM 9.3x with PD Content Pack 9.30.2 environment.

This section includes:

• SM Change -> QC/ALM Defect

To upgrade the synchronization scenario of "SM Changes to QC/ALM Defects" ("Change -> Defect"), refer to "Add the Subform to a Form" and "Add Rule Set Calculations/Validations" in the SM Change -> QC/ALM Defect chapter.

• SM Change -> QC/ALM Requirement

To upgrade the synchronization scenario of "SM Changes to QC/ALM Requirements" ("Change -> Requirement"), refer to Add the Subform to a Form and Add Rule Set Calculations/Validation in the SM Change -> QC/ALM Requirement chapter.

Upgrade from Non-PD Environment to PD Content Pack 9.30.3/SM 9.4x Codeless

This section describes how to upgrade various synchronization scenarios from SM 9.3x non-PD environment to SM 9.3x with PD Content Pack 9.30.3. The following steps are also applicable when upgrading synchronization scenarios from SM 9.3x non-PD environment to SM 9.4x Codeless.

This section includes:

• SM Change -> QC/ALM Defect

To upgrade the synchronization scenario of "SM Changes to QC/ALM Defects" ("Change -> Defect"), refer to "Add the Subform to a Form" and "Add Rule Set Calculations/Validations" in the SM Change - > QC/ALM Defect chapter.

• SM Change -> QC/ALM Requirement

To upgrade the synchronization scenario of "SM Changes to QC/ALM Requirements" ("Change -> Requirement"), refer to Add the Subform to a Form and Add Rule Set Calculations/Validation in the SM Change -> QC/ALM Requirement chapter.

• SM Problem -> QC/ALM Defect

To upgrade the synchronization scenario of "SM Problems to QC/ALM Defects" ("Problem -> Defect"), perform the following tasks:

a. Upgrade SM Customization

Refer to "Specify the External Access Definition on Service Manager", "Add the Subform to a Form" and "Add Rule Set Calculations/Validations" in the SM Problem -> QC/ALM Defect chapter.

b. Upgrade QC/ALM Synchronizer Customization

Refer to "Define Field Mappings" in the SM Problem -> QC/ALM Defect chapter.

• QC/ALM Defect -> SM Problem

To upgrade the synchronization scenario of "QC/ALM Defects to SM Problems" ("Defect -> Problem"), perform the following tasks:

a. Upgrade SM Customization

Refer to "Add the Subform to a Form" in the QC/ALM Defect -> SM Problem chapter.

b. Upgrade QC/ALM Synchronizer Customization

Refer to "Define Field Mappings" in the QC/ALM Defect -> SM Problem chapter.

• SM Problem <-> QC/ALM Defect

To upgrade the synchronization scenario between QC/ALM Defects and SM Problems ("Defect <-> Problem"), perform the following tasks:

a. Upgrade SM Customization

Refer to "Specify the External Access Definition on Service Manager", "Add the Subform to a Form" and "Add Rule Set Calculations/Validations" in the SM Problem -> QC/ALM Defect chapter. b. Upgrade QC/ALM Synchronizer Customization

Refer to "Define Field Mappings" in the SM Problem -> QC/ALM Defect chapter.

Upgrade from 9.3x with PD Content Pack 9.30.2 to 9.30.3/SM 9.4x Codeless/SM 9.4x Hybrid

This section describes how to upgrade various synchronization scenarios from SM 9.3x with PD Content Pack 9.30.2 to SM 9.3x with PD Content Pack 9.30.3. The following steps are also applicable when upgrading synchronization scenarios from SM 9.3x with PD Content Pack 9.30.2 to SM 9.4x Codeless/Hybrid.

This section includes:

• SM Problem -> QC/ALM Defect

To upgrade the synchronization scenario of "SM Problems to QC/ALM Defects" ("Problem -> Defect"), perform the following tasks:

a. Upgrade SM Customization

Refer to "Specify the External Access Definition on Service Manager", "Add the Subform to a Form" and "Add Rule Set Calculations/Validations" in the SM Problem -> QC/ALM Defect chapter.

b. Upgrade QC/ALM Synchronizer Customization

Refer to "Define Field Mappings" in the SM Problem -> QC/ALM Defect chapter.

• QC/ALM Defect -> SM Problem

To upgrade the synchronization scenario of "QC/ALM Defects to SM Problems" ("Defect -> Problem"), perform the following tasks:

a. Upgrade SM Customization

Refer to "Add the Subform to a Form" in the QC/ALM Defect -> SM Problem chapter.

b. Upgrade QC/ALM Synchronizer Customization

Refer to "Define Field Mappings" in the QC/ALM Defect -> SM Problem chapter.

• SM Problem <-> QC/ALM Defect

To upgrade the synchronization scenario between QC/ALM Defects and SM Problems ("Defect <-> Problem"), perform the following tasks:

a. Upgrade SM Customization

Refer to "Specify the External Access Definition on Service Manager", "Add the Subform to a Form" and "Add Rule Set Calculations/Validations" in the SM Problem -> QC/ALM Defect chapter.

b. Upgrade QC/ALM Synchronizer Customization

Refer to "Define Field Mappings" in the SM Problem -> QC/ALM Defect chapter.

Important Notice When Upgrading to SM 9.41 and Above

If you are upgrading various existing synchronization scenarios to HP Service Manager 9.41 Classic/Codeless/Hybrid or above, follow these steps to make the necessary configurations for the new logical name solution introduced by Service Manager 9.41:

- 1. Add the related CI display name in the web service definition.
 - SM Change -> QC/ALM Defect

Add the following fields as described in Step 2 in "Specify the External Access Definition on Service Manager" on page 50.

Field	Caption
affected.itemsm.device.display.name	ServiceName
middle,logical.namesm.device.display.name	ConfigurationItemName
middle,assetssm.device.display.name	AssetsName

SM Problem -> QC/ALM Defect

Add the following fields as described in Step 2 in "Specify the External Access Definition on Service Manager" on page 96.

Field	Caption
affected.itemsm.device.display.name	ServiceName
logical.namesm.device.display.name	AffectedCIName

QC/ALM Defect -> SM Problem

Add the following fields as described in Step 2 in "Specify the External Access Definition on Service Manager" on page 123.

Field	Caption
affected.itemsm.device.display.name	ServiceName
logical.namesm.device.display.name	AffectedCIName

SM Problem <-> QC/ALM Defect

Add the following fields as described in Step 2 in "Specify the External Access Definition on Service Manager" on page 151.

Field	Caption
affected.itemsm.device.display.name	ServiceName
logical.namesm.device.display.name	AffectedCIName

Note: The above fields are for mapping SM Problems to QC/ALM Defects only.

- 2. Refer to "Upgrading the Integration" on page 186 to upgrade your integration.
- 3. Configure the new field mapping.

When retrieving changes and problems from SM, you can get the Service field with a value such as CI1001030. Now you can use the new ServiceName field with a value such as Applications.

4. Change the default value when create new changes and problems.

For example, before you set the Applications value in the Service field to create new changes or problems, you need to search for the CI Identifer in SM by setting the display name to "Applications", instead of using the default "Service" name.

Note: ServiceName and PrimaryCIName do not work in operations such as create or update.

Post-Upgrade Tasks

If you are using the integration solution release 1.00:

A new parameter **Socket timeout (Minutes)** has been introduced since Patch 1. To leverage this parameter, you need to export/import links to make this new parameter available on the Quality Center Synchronizer client.

Perform the following steps:

- Export all links as XML files in the Quality Center Synchronizer client by clicking Link > Export > Link Configuration into XML File....
- 2. Delete each link in the HP Quality Center Synchronizer client by clicking **Link** > **Delete**.
- 3. Restore all links by importing the XML files into the HP Quality Center Synchronizer client by clicking Link > Create From > Link configuration XML File....

Appendix A: Error Messages

This appendix describes the following categories of error messages:

- "Required Fields" on the next page
- "Installation" on page 201
- "Configuration" on page 201
- "Runtime" on page 204
- "XML Validation" on page 208

Required Fields

The following field names are hard-coded.

Field Name	Module	Action	Error Message/Symptom
qcintegration.type	SM Database	Synchronization	No errors in the log and the records failed to be synchronized.
qcintegration.project	SM Database	Synchronization	No errors in the log and the records failed to be synchronized.
QCIntChangeService	SM WSDL Configuration	Generate/Deploy Stub	The stub for Service Manager Change generation failed.
QCIntChange	SM WSDL Configuration	Create a link	System.Web.Services.Protocols.SoapException:java.lang.reflect. InvocationTargetException
ChangeNumber	SM WSDL Configuration	Synchronization	Query: SM ChangeManagement: Can not getRecordIDs. Synchronize: Processing synchronization events failed. Error : java.lang.NoSuchMethodError
Modified	SM WSDL Configuration	Synchronization	Query: SM ChangeManagement: Can not getRecordIDs. Synchronize: Processing synchronization events failed. Error : java.lang.NoSuchMethodError / Query: SM ProblemManagement: Can not getRecordIDs. Synchronize: Processing synchronization events failed. Error : java.lang.NoSuchMethodError
QCIntProblemService	SM WSDL Configuration	Generate/Deploy Stub	Stub for Service Manager Problem generation failed.

Field Name	Module	Action	Error Message/Symptom
QCIntProblem	SM WSDL Configuration	Create a link	System.Web.Services.Protocols.SoapException:java.lang.reflect. InvocationTargetException.
ProblemID	SM WSDL Configuration	Synchronization	Query: SM ProblemManagement: Can not getRecordIDs. Synchronize: Processing synchronization events failed. Error: java.lang.NoSuchMethodError.

Installation

MSG_ ID	Message	Cause	Solution
INS_1	The stub for Service Manager	The Change WSDL	Provide the valid Change WSDL URL. (For example,
	Change generation failed.	URL is not valid.	http://localhost:13080/sc62server/PWS/QCIntChangeService.wsdl).
INS_2	The stub for Service Manager	The Problem WSDL	Provide the valid Problem WSDL URL. (For example,
	Problem generated failed.	URL is not valid.	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 < QCS_Install_Dir>\adapters\lib.

MSG_ ID	Message	Cause	Solution
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 QC Endpoint 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 the Events tab.
CFG_ 4	Missing connection parameter: UserName.	'User name' is empty.	Enter a user name in the Connectivity tab.
CFG_ 5	Missing connection parameter: Service URL.	'Service URL' is empty.	Enter a Service URL in the Connectivity tab.
CFG_ 6	Missing connection parameter: QC Project.	'QC Project' is empty.	Enter a QC Project name in the Connectivity tab.
CFG_ 7	Missing connection parameter: Service URL.	'Service URL' is empty.	Enter a Service URL in the Connectivity tab.
CFG_ 8	Connection parameter: Configuration File Path is not valid.	File path is invalid and the specified file does not exist.	Enter a valid configuration file path name in the Connectivity tab (or leave it empty).

MSG_ ID	Message	Cause	Solution
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 it 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.
CFG_ 11	ERROR #1- adapter.CONNECTION_FAILURE : Failed to connect to endpoint of type SM ChangeManagement. Error: com.hp.qc.synchronizer.adapters.exceptions. 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 QCIntProblemService].wsd</port></sm>

MSG_ ID	Message	Cause	Solution
CFG_ 12	Retry times must be an integer between 0 and 3. (0 means disabled).	The value for the parameter Retries on Locked Record in the Advanced tab is out of range (0~3).	Enter an integer (0 ~ 3) for Retries on Locked Record in the Advanced tab.
CFG_ 13	Retry interval must be an integer between 1 and 10.	The value for parameter Retry Interval(Seconds) in the Advanced tab is out of range (1 to 10).	Enter an integer (1 to 10) for Retry Interval (Seconds) in the Advanced tab.
CFG_ 14	Socket timeout must be an integer between 0 and 120. (0 means default timeout).	The value for parameter: Socket Timeout (Minutes) in the Advanced tab is out of range (0 to 120).	Enter an integer (0 to 120) for Socket Timeout (Minutes) in the Advanced tab.

Runtime

MSG_ ID	Message	Cause	Solution
RUN_ 1	Required field <field Name> can not be empty or SPACE filled.</field 	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: Resource Unavailable	Synchronize updates from QC side to locked change/problem records.	Close the locked records in SM side.

MSG_ ID	Message	Cause	Solution
RUN_ 2	Error when reading web service response from SM: Not authorized	 There are two possible causes: Insufficient rights for the SM Integration user when creating/updating defect/requirement in QC or running synchronization to create/update the corresponding change/problem. Maximum active logins for integration account is exceeded. 	 Check the rights of integration account. Check and make sure that Unlimited session in the Security tab is selected.
RUN_ 3	Error 23scxmlapi(23) - XML DOM exception caught - code 5 msg An invalid or illegal XML character is specified	Synchronization is performed with an illegal WSDL caption.	Correct the 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 the record manually or run a full synchronization to run all missing updates.
RUN_ 6	Cannot get field for <field Name>.</field 	SM adapter cannot get a field.	Ensure that the stub jar has been generated correctly.

MSG_ ID	Message	Cause	Solution
RUN_ 7	Value cannot be reached for <field name="">.</field>	SM adapter cannot find this field from the stub class.	Ensure that the stub jar has been generated correctly.
RUN_ 8	Mapping error, no such property <property name=""> defined in type <type name>.</type </property>	SM adapter cannot find this property.	Ensure that the stub jar has been generated correctly.
RUN_ 9	Error during setting value for key <key name=""> with value <value>.</value></key>	Dynamic model cannot find this key.	Ensure that the stub jar has been generated correctly.
RUN_ 10	Mapping file <file path=""> cannot be found!</file>	The configuration file isn't found.	Please check the configuration file.
RUN_ 10	<module builder="" class<br="">name> cannot be created because of <message>.</message></module>	SM adapter cannot load a specified class.	Ensure that the stub jar has been generated correctly.
RUN_ 11	Can not convert to <target class name> from value <value>.</value></target 	Value cannot be converted to target type.	Ensure that the WSDL does not expose non- supported data types.
RUN_ 12	Exception when getting SM response, return code: <return code="">.</return>	Problem with SM communication.	Refer to references for error messages.

MSG_ ID	Message	Cause	Solution
RUN_ 13	Error when reading web service response from SM, record is locked [changeID= <recordid>], message=<message>.</message></recordid>	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.</file </record></field 	The SOAP field data type in the WSDL is not correct. If the field is of the Number type, the value in the database is out of the range 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 value of 2,147,483,648 (2^31), it will cause this error when reading the record through the web service interface.	If this field has a Number type, choose DecimalType in the WSDL. Otherwise select the correct SOAP type.
RUN_ 15	Unable to create envelope from given source:	The name of a structure field in Service Manager may have non-English characters.	Use English characters in the name of a structure field.
RUN_ 16	QC: findRequirementByld: Failed getting requirement with id: <id> Failed to update, record was not found or deleted on target null</id>	Deleting or removing a record may result in this problem, because incremental synchronization will fail to find the record.	Restore this record or just run a "Full Synchronization" to remove this mapping relationship established for the record before.
RUN_ 17	Invalid byte 2 of 3-byte UTF-8 sequence.	There are some special non-English characters in the values of fields.	Update the WSDL definition for this field in Service Manager by leaving the "Type" field bank instead of specifying "StringType" for this field.

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 a 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 a 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 two module elements in the mapping file.	Make sure the mapping element has only 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 a module is not problem or change .	The name attribute of a module element should be change or problem .

MSG_ ID	Error Message	Cause	Solution
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 'Unknown' 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 'Unknown' of attribute 'type' on element 'field' is not valid with respect to its type, 'FieldType'.	The field element has a wrong type attribute.	The type attribute of a 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'.	The field element has a wrong readonly attribute.	The readonly attribute of a field element 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'.	The field element has a wrong required attribute.	The required attribute of a field element should be mandatory , optional or recommended .
XML_ 9	Failed to validate the configuration file: cvc-complex-type.4: Attribute 'type' must appear on element 'field'.	The field element has no type attribute.	The type attribute must be defined in each field element.

MSG_ ID	Error Message	Cause	Solution
XML_ 10	Failed to validate the configuration file: cvc-complex-type.4: Attribute 'name' must appear on element 'field'.	The field element has no name attribute.	Define a name attribute in the 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.	The field element has more than one child element items .	Define only one items element 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.	The items element has no child element item .	Add item elements in each items element.
XML_ 13	Failed to validate the configuration file: cvc-complex-type.4: Attribute 'value' must appear on element 'item'.	The item element has no value attribute.	Define a 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.	The item element has no text value.	Define a text value for each item element.

MSG_ ID	Error Message	Cause	Solution
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'.	The value attribute has an empty value.	Define a value for the value attribute in each 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'.	The name attribute has an empty value.	Define a value for the name attribute of each 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'.	The field element has a type of Attachment .	Remove the 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'.	The field element has an incorrect length attribute value.	Correct the value of the field element.

Send Documentation Feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

Feedback on Installation and Administration Guide (Defects and Requirements Exchange with HP Service Manager and HP Application Lifecycle Management 1.03)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to ovdoc-itsm@hp.com.

We appreciate your feedback!



