

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

Software Version: 1.03 for Process Designer (PD) Content Pack 9.30.2 and 9.30.3

For the supported Windows® operating system

## **Installation and Administration Guide**

Document Release Date: April 2015

Software Release Date: April 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 Online website 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 sign in. Many also require a support contract. To register for an HP Passport ID, go to:

<http://h20229.www2.hp.com/passport-registration.html>

To find more information about access levels, go to:

[http://h20230.www2.hp.com/new\\_access\\_levels.jsp](http://h20230.www2.hp.com/new_access_levels.jsp)

**HP Software Solutions Now** accesses the HPSW Solution and Integration 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 <http://h20230.www2.hp.com/sc/solutions/index.jsp>

# Contents

<b>Chapter 1: Introduction</b>	<b>9</b>
Document Purpose	9
Synchronization Concepts	10
References	11
<b>Chapter 2: Planning the Deployment</b>	<b>12</b>
Supported Products and Platforms	12
Deployment Scenarios	13
Data Types	13
Deployment Tasks	13
Release Package	14
<b>Chapter 3: Installing and Configuring QC/ALM Synchronizer</b>	<b>15</b>
Installing QC/ALM Synchronizer	15
Installation Requirements	16
Deploying the Adapters	16
Generating/Deploying the Stub	17
Copying SM Adapter Configuration Files	17
Editing the Configuration Files	18
Module Types	18
Field Types	18
Requirements	19
SM Field Type and Definition Rule	20
SM Change Management Example	21
SM Problem Management Example	21
<b>Chapter 4: Configuring Links in QC/ALM Synchronizer</b>	<b>24</b>
Create a Link	25
Advanced Parameters	28
QC/ALM Field <-> SM Field Mappings	29

Creating Mappings .....	29
General Mapping Requirements .....	30
Matching Types .....	31
List Value Mappings .....	35
Constant -> SM Field Mappings .....	35
<b>Chapter 5: Integration Account .....</b>	<b>36</b>
Creating an SM Integration Account .....	36
Create a Contact Record .....	36
Create a Profile Record .....	36
Create a SecRole Record .....	39
Create an Operator Record .....	40
Creating a QC/ALM Integration Account .....	42
<b>Chapter 6: SM Change -&gt; QC/ALM Defect .....</b>	<b>49</b>
Customizing Service Manager/ServiceCenter for Change Management .....	49
Add Fields .....	50
Specify the External Access Definition on Service Manager .....	50
Specify the External Access Definition on ServiceCenter .....	52
Create a Subform .....	53
Add the Subform to a Form .....	55
Add Format Control Calculations/Validations .....	57
Add Rule Set Calculations/Validations .....	59
Copy an Existing Workflow .....	59
Associate an Existing Change Category with the New Workflow .....	60
Create New Rule Set for Initialization and Validation .....	61
Associate the New Workflow with the New Rule Set .....	64
Customizing the QC/ALM Defects Module .....	66
On QC 10 or Earlier .....	66
Add Fields .....	66
Add Tabs .....	67
Add Fields to Tabs .....	69
Verify .....	70
On ALM 11 .....	72
Add fields .....	72

Verify .....	73
Configuring Links in QC/ALM Synchronizer .....	75
Specify Endpoints / Type of Link .....	75
Define Field Mappings .....	76
QC/ALM Field <-> SM Field .....	77
Define Events .....	80
Test the Link .....	80
<b>Chapter 7: SM Change -&gt; QC/ALM Requirement .....</b>	<b>83</b>
Customizing Service Manager/ServiceCenter for Change Management .....	83
Customizing the QC/ALM Requirements Module .....	83
On QC 10 or Earlier .....	83
Add Fields .....	84
Add Tabs .....	85
Add Fields to Tabs .....	87
Create the SM Incoming Changes Folder .....	87
On ALM 11 .....	88
Add Fields .....	88
Create the SM Incoming Changes Folder .....	89
Configuring Links in QC/ALM Synchronizer .....	90
Specify Endpoints / Type of Link .....	90
Define Field Mappings .....	92
QC/ALM Field <-> SM Field .....	93
Define Events .....	94
Test the Link .....	94
<b>Chapter 8: SM Problem -&gt; QC/ALM Defect .....</b>	<b>97</b>
Customizing Service Manager/ServiceCenter for Problem Management .....	97
Add Fields .....	98
Specify the External Access Definition on Service Manager .....	98
Specify the External Access Definition on ServiceCenter .....	103
Create a Subform .....	105
Add the Subform to a Form .....	108
Add Format Control Calculations/Validations .....	109
Add Rule Set Calculations/Validations .....	110

Copy an Existing Workflow .....	111
Associate an Existing Problem Category with the New Workflow .....	111
Create New Rule Set for Initialization and Validation .....	112
Associate the New Workflow with the New Rule Set .....	114
Customizing the QC/ALM Defects Module .....	116
On QC 10 and Earlier .....	116
Add Fields .....	116
Add Tabs .....	117
Add Fields to Tabs .....	118
On ALM 11 .....	120
Add Fields .....	120
Configuring Links in QC/ALM Synchronizer .....	122
Specify Endpoints / Type of Link .....	122
Define Field Mappings .....	122
Define Events .....	123
Test the Link .....	124
<b>Chapter 9: QC/ALM Defect -&gt; SM Problem .....</b>	<b>126</b>
Customizing Service Manager/ServiceCenter for Problem Management .....	126
Add Fields .....	127
Specify the External Access Definition on Service Manager .....	127
Specify the External Access Definition on ServiceCenter .....	131
Create a Subform .....	136
Add the Subform to a Form .....	137
Customizing the QC/ALM Defects Module .....	139
On QC 10 and Earlier .....	139
Add Fields .....	139
Add Tabs .....	140
Add Fields to Tabs .....	141
Create a View .....	143
Verify .....	144
On ALM 11 .....	144
Add Fields .....	144
Create a View .....	146
Verify .....	147

Configuring Links in QC/ALM Synchronizer .....	147
Specify Endpoints / Type of Link .....	147
Define Filters .....	148
Define Field Mappings .....	148
QC/ALM Field <-> SM Field .....	151
Constants -> SM Fields .....	152
Define Events .....	154
Test the Link .....	154
 Chapter 10: SM Problem <-> QC/ALM Defect .....	157
Customizing Service Manager/ServiceCenter for Problem Management .....	157
Add Fields .....	158
Specify the External Access Definition on Service Manager .....	158
Specify the External Access Definition on ServiceCenter .....	163
Create a Subform .....	165
Add the Subform to a Form .....	168
Add Format Control Calculations/Validations .....	169
Add Rule Set Calculations/Validations .....	171
Copy an Existing Workflow .....	171
Associate an Existing Problem Category with the New Workflow .....	172
Create New Rule Set for Initialization and Validation .....	173
Associate the New Workflow with the New Rule Set .....	175
Customizing the QC/ALM Defects Module .....	177
On QC 10 and Earlier .....	177
Add Fields .....	177
Add Tabs .....	178
Add Fields to Tabs .....	180
Create a View .....	182
Verify .....	182
On ALM 11 .....	183
Add Fields .....	183
Create a View .....	185
Verify .....	185
Configuring Links in QC/ALM Synchronizer .....	186
Specify Endpoints / Type of Link .....	186
Define Filters .....	187

Define Field Mappings .....	187
Define Events .....	190
Test the Link .....	191
<b>Chapter 11: Upgrading the Integration .....</b>	<b>195</b>
Upgrading to the Latest Release .....	195
Back up Jar Files and Links .....	195
Deploy the Latest Adapters .....	195
Upgrade for Various Synchronization Scenarios .....	196
Upgrade in Non-PD Environment .....	196
SM Change -> QC/ALM Defect .....	197
SM Change -> QC/ALM Requirement .....	198
SM Problem -> QC/ALM Defect .....	199
QC/ALM Defect -> SM Problem .....	199
SM Problem <-> QC/ALM Defect .....	202
Upgrade from Non-PD Environment to PD Content Pack 9.30.2 (for SM 9.3x only) .....	202
Upgrade from Non-PD Environment to PD Content Pack 9.30.3/SM 9.4x Codeless .....	202
Upgrade from 9.3x with PD Content Pack 9.30.2 to 9.30.3/SM 9.4x Codeless .....	204
Post-Upgrade Tasks .....	205
<b>Appendix A: Error Messages .....</b>	<b>206</b>
Required Fields .....	207
Installation .....	208
Configuration .....	208
Runtime .....	211
XML Validation .....	215
<b>Send Documentation Feedback .....</b>	<b>219</b>



# 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 and its "isKnownError" flag is true.

- **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" on the next page](#)
- ["Data Types" on the next page](#)
- ["Deployment Tasks" on the next page](#)
- ["Release Package" on page 14](#)

### Supported Products and Platforms

Supported products are shown in the following table.

Supported Product	Version
HP Quality Center Synchronizer / HP ALM Synchronizer	1.2, 1.3, 1.4, 1.5, and 1.6
HP Quality Center / HP Application Lifecycle Management	9.2 patch 4 and above, 10, 11, 11.5, and 12
HP Service Manager	7.01, 7.02, 7.10, 9.20 patch 1, 9.3x and 9.40
HP ServiceCenter	6.2.2 and above

- In this integration solution, supported platforms for Quality Center Synchronizer or ALM Synchronizer include:
- Microsoft Windows 2000 with Service Pack 4 (32bit)
- Microsoft Windows XP with Service Pack 2 (32bit)
- Microsoft Windows 2003 Server with Service Pack 2 (32bit)

**Note:** When used with ALM 11.00, ALM Synchronizer only supports Microsoft Windows XP with Service Pack 2 (32bit).

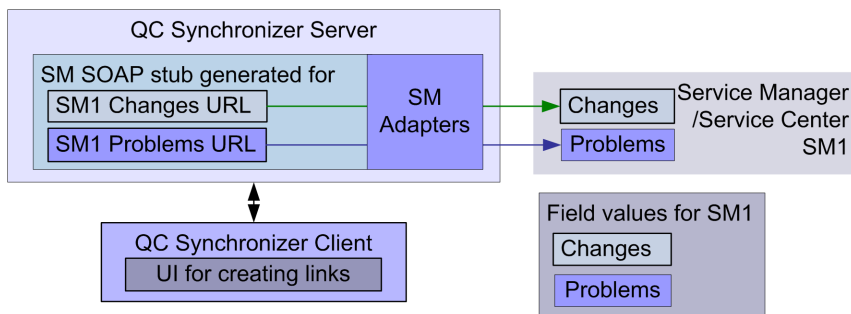
For information about supported platforms of Service Manager, and Quality Center/ALM, refer to their product support matrix.

## 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 ALM Synchronizer 1.4.

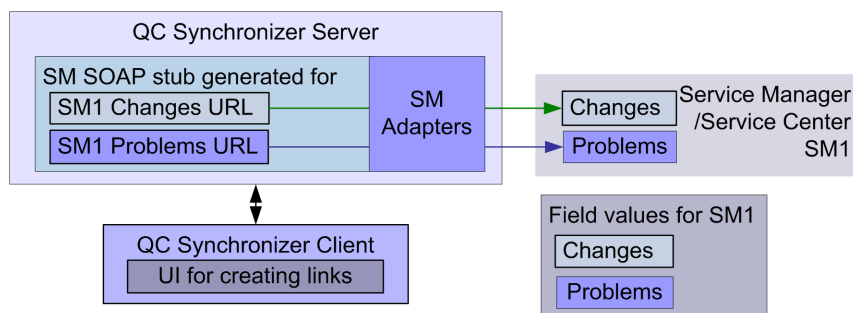
## Chapter 3: Installing and Configuring QC/ALM Synchronizer

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
```

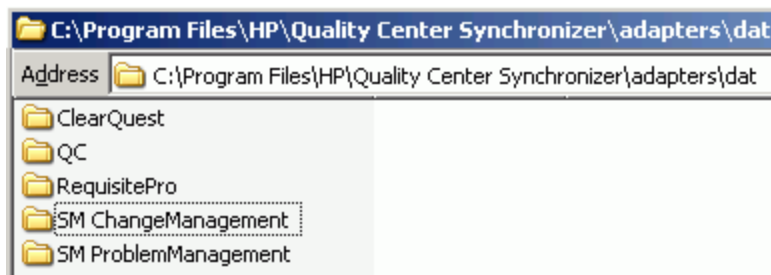
3. Run the script build.bat from the operating system's command prompt (check the console output for errors). The stub <Release\_Package>\build\sm-adapter-ws-client.jar is generated.
4. Copy the stub to the <QCS\_Install\_Dir>\adapters\lib directory.

## 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).



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 <b>Caption</b> enabled in the SM/SC WSDL. This field is required.

**Field XML Element Specification, continued**

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.

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 Field Type 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
Date/time	Date/time	Date	DateTimeType	Optional. Permitted type is "Date".
Number	Decimal	Decimal Text	<Empty> DecimalType IntType	Optional. Permitted type is "Number".
Logical	Boolean	Check Box Radio Button	<Empty> BooleanType	Optional. Permitted type is "String".
Character	Text	Text TextArea Combo Box Comfill	<Empty> StringType	Optional. Permitted types include "String" and "Single_Value_List".

**SM Field Type and Definition Rules, continued**

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>	Optional. Permitted types include "String" and "Single_Value_List".
Array	Array	Combo Box Comfill Text	<Empty>	Required. Permitted type is "Multi_Value_List".

## 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/SMQCIIntegration/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/SMQCIIntegration/config">

  <itg:module name="problem">
    <itg:field name="Status" type="Single_Value_List" required="mandatory">
```

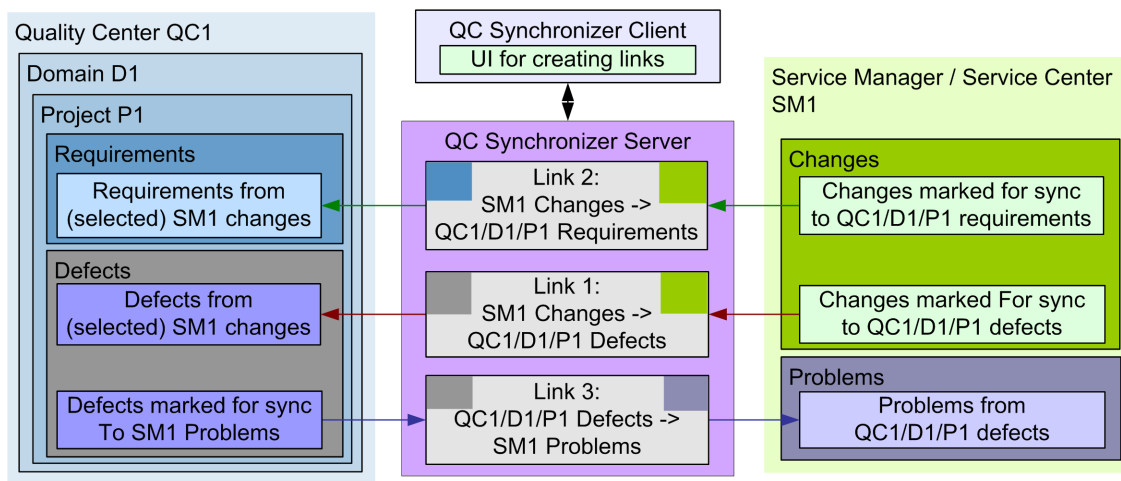
```
<itg:items>
  <itg:item value="Open">Open</itg:item>
  <itg:item value="Accepted">Accepted</itg:item>
  <itg:item value="Work In Progress">Work In Progress</itg:item>
  <itg:item value="Pending Vendor">Pending Vendor</itg:item>
  <itg:item value="Pending User">Pending User</itg:item>
  <itg:item value="Rejected">Rejected</itg:item>
  <itg:item value="Deferred">Deferred</itg:item>
</itg:items>
</itg:field>
<itg:field name="AssignmentGroup" type="Single_Value_List" required="mandatory">
  <itg:items>
    <itg:item value="Application">Application</itg:item>
    <itg:item value="Network">Network</itg:item>
  </itg:items>
</itg:field>
<itg:field name="Service" type="Single_Value_List" required="mandatory">
  <itg:items>
    <itg:item value="Applications">Applications</itg:item>
    <itg:item value="Service Management">Service Management</itg:item>
  </itg:items>
</itg:field>
<itg:field name="Title" type="String" required="mandatory" length="50"/>
  <itg:field name="Description" type="String" required="mandatory"/>
<itg:field name="Area" type="Single_Value_List" required="mandatory">
  <itg:items>
    <itg:item value="data">data</itg:item>
  </itg:items>
</itg:field>
<itg:field name="Subarea" type="Single_Value_List" required="mandatory">
  <itg:items>
```

```
<itg:item value="data or file corrupted">data or file
corrupted</itg:item>
</itg:items>
</itg:field>
<itg:field name="Impact" type="Single_Value_List" readonly="false"
required="mandatory">
<itg:items>
<itg:item value="1">1 - Enterprise</itg:item>
<itg:item value="2">2 - Site/Dept</itg:item>
<itg:item value="3">3 - Multiple Users</itg:item>
<itg:item value="4">4 - User</itg:item>
</itg:items>
</itg:field>
<itg: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 148](#)). 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

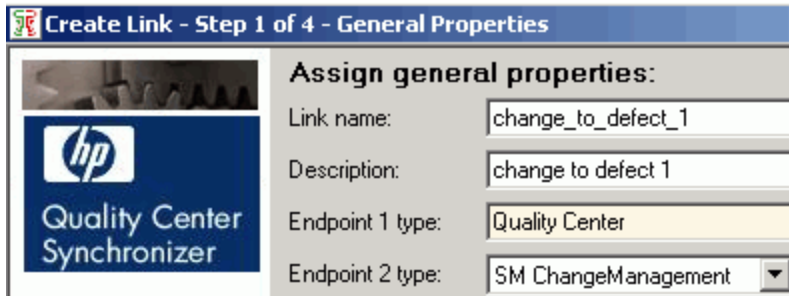
The following table summarizes the properties required in the wizard. Have this data available before starting the wizard.

**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.

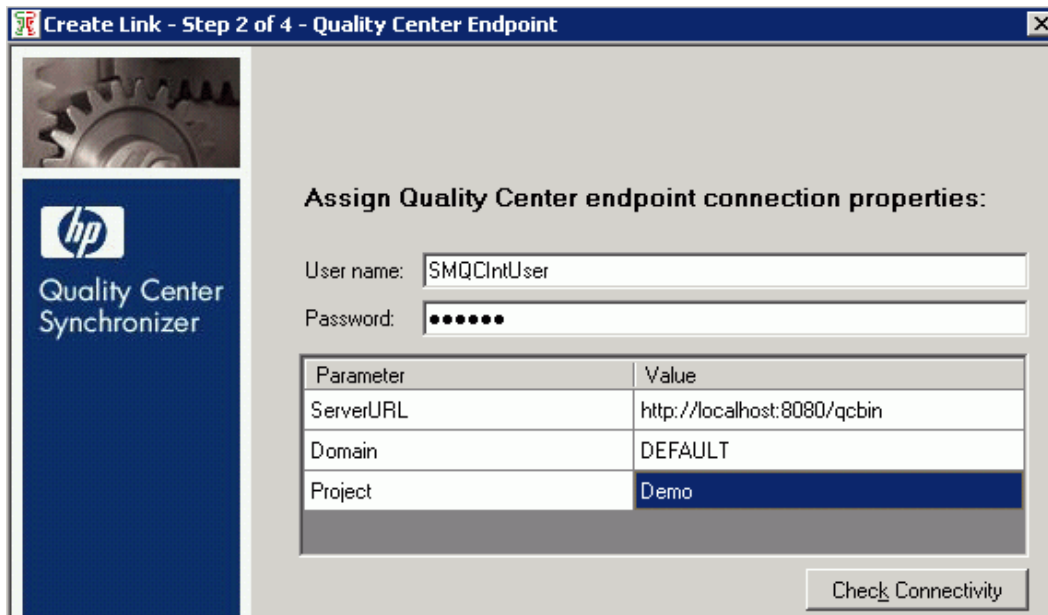
End Point	Parameter	Requirements
QC	Username	
QC	Password	
QC	Server URL	
QC	Domain	
QC	Project	
SM	User name	
SM	Password	
SM	Service URL	<code>http://&lt;service_manager_host&gt;: &lt;port&gt;/sc62server/PWS/QCIntChangeService.wsdl</code> or <code>http://&lt;service_manager_host&gt;: &lt;port&gt;/sc62server/PWS/QCIntProblemService.wsdl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

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).



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



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

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).

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

**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

Available entity type pairs: Defect(QC) - Change as Defect(SM ChangeManag)

9. Click **Save**. The link is created.

10. Modify required settings on the **Connectivity** tab.

General Connectivity Scheduling Filters Events Field Mapping Advanced

QC

User name: SMQCIntUser

Password: .....

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

SM ChangeManagement

User name: SMQCIntUser

Password: .....

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

## Advanced Parameters

Advanced parameters are shown on the Advanced tab.

SM ChangeManagement Adapter Settings

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*.

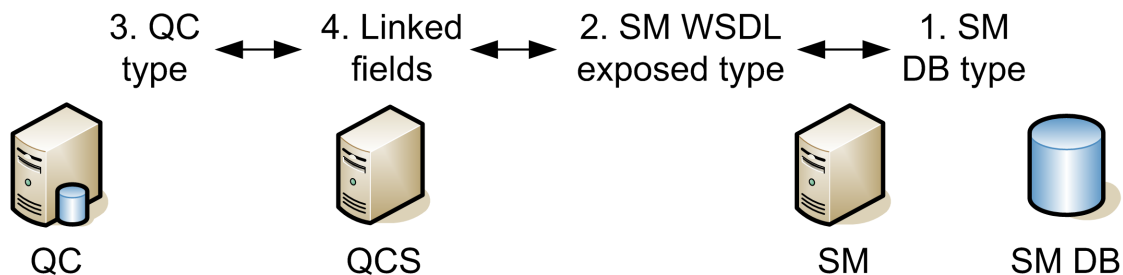
- **Socket Timeout (Minutes)**  
Socket connection will be established during synchronization. If there are many records matching the filter in the SM endpoint, retrieving list operation will cost some time, which might cause timeout of socket connection. This parameter is used to define the socket timeout. 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.



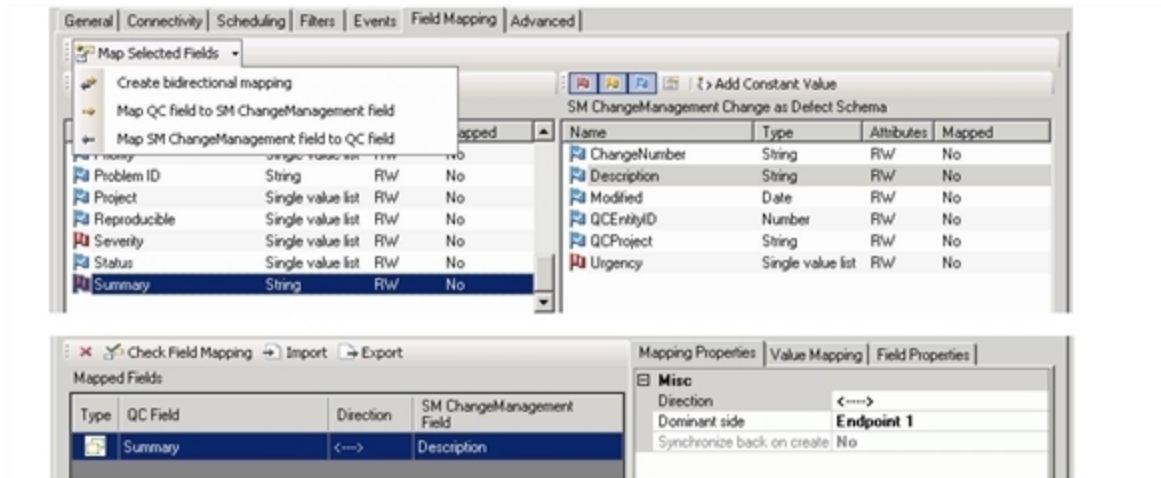
For examples of field mappings, see:

- SM Change to QC/ALM Defect, ["QC/ALM Field <-> SM Field" on page 77](#).
- SM Change to QC/ALM Requirement, ["QC/ALM Field <-> SM Field" on page 93](#).
- QC/ALM Defect to SM Problem, ["QC/ALM Field <-> SM Field" on page 151](#).

## Creating Mappings

To map fields:

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



## 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 <sup>1</sup>	Decimal or Text	Number	Decimal
String	String	<->	String	BooleanType	Check Box or Radio Button	Logical	Boolean
Date <sup>2</sup>	Date	<->	Date	DateTimeType (required)	Date	Date/time	Date/time
String	String	<->	String <sup>3</sup>	StringType	Text, TextArea, Combo Box or Comfill	Character	Text
Memo	Memo/ String	<->	String <sup>4</sup>	StringType	Text, TextArea, Combo Box or Comfill	Character	Text
User List <sup>5</sup>	User List	->	String <sup>6</sup>	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 <sup>7</sup>	StringType	Text, TextArea, Combo Box or Comfill	Character	Text
String	String	<->	String	StringType	TextArea	Array <sup>8</sup>	Array <sup>9</sup>
Memo	Memo String	<->	String	StringType	TextArea	Array <sup>10</sup>	Array <sup>11</sup>
Lookup List	Single-value List	<->	String	StringType	TextArea	Array <sup>12</sup>	Array <sup>13</sup>
User List <sup>14</sup>	User List	<->	String	StringType	TextArea	Array <sup>15</sup>	Array <sup>16</sup>
Lookup List	Multi-value List	<->	Multi-value List	StringType	Text, Comfill or Combo Box	Array <sup>17</sup>	Array <sup>18</sup>
Attachment <sup>19</sup>	Attachment	<->	Attachment	Attachment	Attachment	Image	Image

**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*.

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

<sup>2</sup>QC data only supports Yr/Mo/Dt.

<sup>3</sup>It is recommended to leave this field blank. Otherwise "Invalid byte 2 of 3-byte UTF-8 sequence" might occur if certain I18N characters are synchronized.

<sup>4</sup>See footnote 3.



<sup>5</sup>Write to the QC field `User_List` only if SM has exactly the same users (including logins, names, etc.) as QC. An incorrect entry can cause serious problems in QC. You can read from QC `User_List` field and write to SM `String` type field only if the field in SM is NOT a field with SM logins.

<sup>6</sup>See footnote 3.

<sup>7</sup>See footnote 3.

<sup>8</sup>Only an array of characters is supported.

<sup>9</sup>See footnote 8.

<sup>10</sup>See footnote 8.

<sup>11</sup>See footnote 8.

<sup>12</sup>See footnote 8.

<sup>13</sup>See footnote 8.

<sup>14</sup>See footnote 5.

<sup>15</sup>See footnote 8.

<sup>16</sup>See footnote 8.

<sup>17</sup>See footnote 8.

<sup>18</sup>See footnote 8.

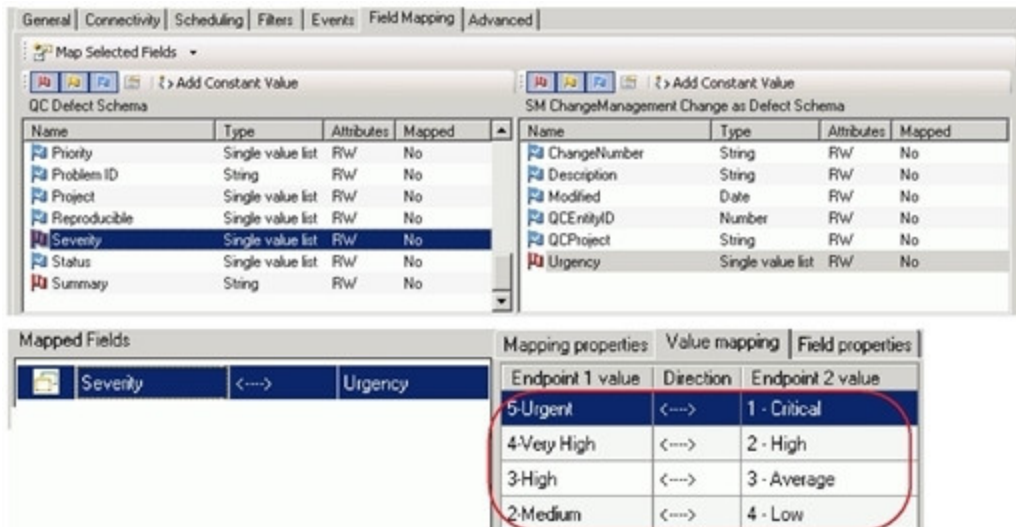
<sup>19</sup> The SM-QC/ALM integration does not support synchronization of an attachment with OKB size. Consider the following guidelines when mapping attachment fields:

- 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.

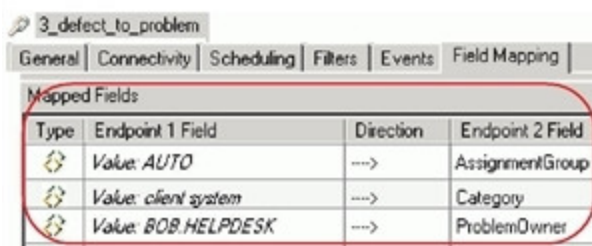
Some list fields also require mapping of available values (as shown in the following figure).



## Constant -> SM Field Mappings

For examples of constant -> SM field mappings, see ["Constants -> SM Fields" on page 152](#).

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



## 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>
Contact Information	Full Name	<Administrator's full name>

### 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.

### Operator Record Parameters

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>	The contact created in the previous section.
4	Security	Unlimited Sessions	Yes	Check Box
5	Security	Password	<Your password>	
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 Defects ("Change -> Requirement" and "Change -> Defect").
10	General/Application Profiles	Change Profiles	CMPProfile_QCInt	



**Operator Record Parameters, continued**

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

**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
	Add Requirement  Modify Requirement	<input checked="" type="checkbox"/> Add Requirement <input checked="" type="checkbox"/> Modify Requirement <input type="checkbox"/> Delete Requirement <input type="checkbox"/> Add Tests To Coverage <input type="checkbox"/> Remove Tests From Coverage <input type="checkbox"/> Add Requirement Traceability <input type="checkbox"/> Modify Requirement Traceability <input type="checkbox"/> Remove Requirement Traceability <input type="checkbox"/> Risk-Based Quality Management	Select the following options: <ul style="list-style-type: none"> <li>Coverage Level               <ul style="list-style-type: none"> <li>Create</li> <li>Update</li> </ul> </li> <li>Requirement               <ul style="list-style-type: none"> <li>Create</li> <li>Update</li> </ul> </li> <li>Risk-Based Quality Management               <ul style="list-style-type: none"> <li>Assess Business Criticality</li> <li>Assess Failure Probability</li> <li>Assess Functional Complexity</li> </ul> </li> </ul>

◦ Change -> Defect

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

Defects	Add Defects/ Modify Defects	<input checked="" type="checkbox"/> Add Defect <input checked="" type="checkbox"/> Modify Defect <input type="checkbox"/> Delete Defect <input type="checkbox"/> Add Defect Link <input type="checkbox"/> Modify Defect Link <input type="checkbox"/> Remove Defect Link	Select the following options: <ul style="list-style-type: none"> <li>Defect               <ul style="list-style-type: none"> <li>Create</li> </ul> </li> <li>Update</li> </ul>
---------	--------------------------------	---	--

◦ Problem -> Defect

Subtab	Permission	QC Setting	ALM Setting
Defects	Add Defects/ Modify Defects	<input checked="" type="checkbox"/> Add Defect <input checked="" type="checkbox"/> Modify Defect <input type="checkbox"/> Delete Defect <input type="checkbox"/> Add Defect Link <input type="checkbox"/> Modify Defect Link <input type="checkbox"/> Remove Defect Link	Select the following options: <ul style="list-style-type: none"> <li>Defect               <ul style="list-style-type: none"> <li>Create</li> </ul> </li> <li>Update</li> </ul>

◦ Problem <- Defect

Subtab	Permission	QC Setting	ALM Setting
Defects	Add Defects/ Modify Defects	<input type="checkbox"/> Add Defect <input checked="" type="checkbox"/> Modify Defect <input type="checkbox"/> Delete Defect <input type="checkbox"/> Add Defect Link <input type="checkbox"/> Modify Defect Link <input type="checkbox"/> Remove Defect Link	Select the following options: <ul style="list-style-type: none"> <li>Defect               <ul style="list-style-type: none"> <li>Create</li> </ul> </li> <li>Update</li> </ul>

Administration	<p>Add Public Favorite Views</p> <p>Modify Public Favorite Views</p> <p>Delete Public Favorite Views</p> <p>Add Private Favorite Views</p> <p>Modify Private Favorite Views</p> <p>Delete Private Favorite Views</p>	<div> <input checked="" type="checkbox"/> Add Public Favorite Views         </div> <div> <input checked="" type="checkbox"/> Modify Public Favorite Views         </div> <div> <input checked="" type="checkbox"/> Delete Public Favorite Views         </div> <div> <input checked="" type="checkbox"/> Add Private Favorite Views         </div> <div> <input checked="" type="checkbox"/> Modify Private Favorite Views         </div> <div> <input checked="" type="checkbox"/> Delete Private Favorite Views         </div>	<p>Select the following options:</p> <ul style="list-style-type: none"> <li>• Add Public Favorite View Folders</li> <li>• Add Public Favorite Views</li> <li>• Delete Public Favorite View Folders</li> <li>• Delete Public Favorite Views</li> <li>• Manage Private Favorite Views</li> <li>• Manage Project Planning and Tracking</li> <li>• Modify Public Favorite View Folders</li> <li>• Modify Public Favorite Views</li> </ul>
----------------	--	--	---

◦ Problem <- Defect

		QC setting	ALM setting
Defects	Add Defects/ Modify Defects	<div> <input checked="" type="checkbox"/> Add Defect         </div> <div> <input checked="" type="checkbox"/> Modify Defect         </div> <div> <input type="checkbox"/> Delete Defect         </div> <div> <input type="checkbox"/> Add Defect Link         </div> <div> <input type="checkbox"/> Modify Defect Link         </div> <div> <input type="checkbox"/> Remove Defect Link         </div>	<p>Select the following options:</p> <ul style="list-style-type: none"> <li>• Defect             <ul style="list-style-type: none"> <li>• Create</li> <li>• Update</li> </ul> </li> </ul>

Administration	<p>Add Public Favorite View</p> <p>Modify Public Favorite Views</p> <p>Delete Public Favorite Views</p> <p>Add Private Favorite Views</p> <p>Modify Private Favorite Views</p> <p>Delete Private Favorite Views</p>	<div> <input checked="" type="checkbox"/> <b>Add Public Favorite Views</b> </div> <div> <input checked="" type="checkbox"/> <b>Modify Public Favorite Views</b> </div> <div> <input checked="" type="checkbox"/> <b>Delete Public Favorite Views</b> </div> <div> <input checked="" type="checkbox"/> <b>Add Private Favorite Views</b> </div> <div> <input checked="" type="checkbox"/> <b>Modify Private Favorite Views</b> </div> <div> <input checked="" type="checkbox"/> <b>Delete Private Favorite Views</b> </div>	<p>Select the following options:</p> <ul style="list-style-type: none"> <li>• Add Public Favorite View Folders</li> <li>• Add Public Favorite Views</li> <li>• Delete Public Favorite View Folders</li> <li>• Delete Public Favorite Views</li> <li>• Manage Private Favorite Views</li> <li>• Manage Project Planning and Tracking</li> <li>• Modify Public Favorite View Folders</li> <li>• Modify Public Favorite Views</li> </ul>
----------------	---	--	---

- 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

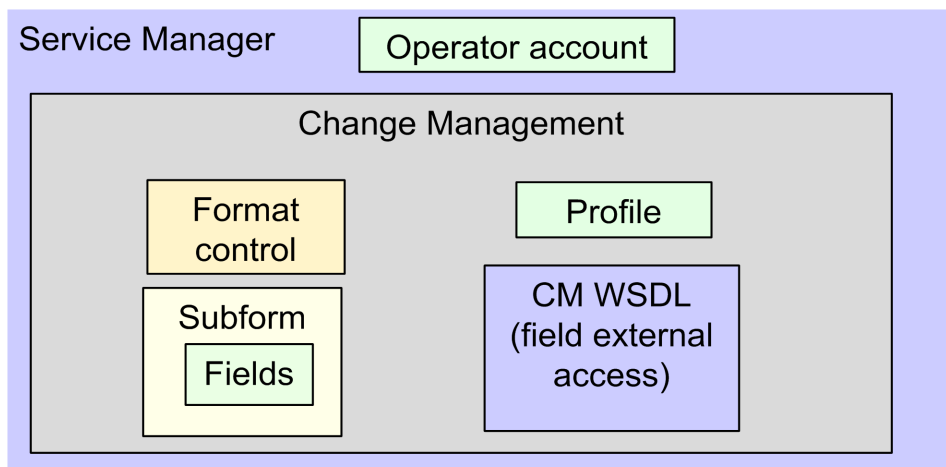
This chapter describes how to synchronize SM Changes with QC/ALM Defects ("Change -> Defect").

This chapter includes:

- ["Customizing Service Manager/ServiceCenter for Change Management" below](#)
- ["Customizing the QC/ALM Defects Module" on page 66](#)
- ["Configuring Links in QC/ALM Synchronizer" on page 75](#)

### Customizing Service Manager/ServiceCenter for Change Management

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



To customize Service Manager/ServiceCenter 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](#) or ["Specify the External Access Definition on ServiceCenter" on page 52](#)
3. ["Create a Subform" on page 53](#)

4. ["Add the Subform to a Form" on page 55](#)
5. ["Add Format Control Calculations/Validations" on page 57](#)

## 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.

Field	Type	
	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:

1. Create a custom External Access Definition `QCIntChangeService` by clicking **Tailoring > WSDL configuration** on Service Manager 7.0x; or clicking **Tailoring > Web Services > WSDL 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

QCIntChange

External Access Definition

Service Name

QCIntChangeService

Name:

cm3r

Object Name:

QCIntChange

Allowed Actions

Expressions

Fields

Allowed Actions	Action Names	Action Type
save	Update	

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

2. Enable required fields in the web service.

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

object.name  
Change  
QCIntChange

**External Access Definition**

Service Name:   
Name:  Object Name:

Allowed Actions Expressions Fields

Field	Caption	Type
header,number	Change <b>Number</b>	StringType
qcintegration.id	QCEntityID	IntType
severity	Urgency	StringType
sysmodtime	Modified	DateTimeType
header,reason	Reason	
description.structure,desc...	Description	
middle,logical.name	ConfigurationItem	
header,risk.assessment	RiskAssessment	
header,coordinator	Coordinator	
header,requested.by	RequestedBy	
header,priority.code	Priority	

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

## Specify the External Access Definition on ServiceCenter

All fields of the ServiceCenter Change entity or Problem entity can be exposed in Web services by modifying their WSDL configuration. In ServiceCenter, you can modify the WSDL configuration by changing the Web Services API properties of the fields in table definition.

**Note:** Restart the ServiceCenter server whenever you make changes to a WSDL configuration.

To specify the External Access Definition on ServiceCenter:

1. Click **System Definition > Tables > cm3r > Fields and keys definitions for cm3r table** and modify the settings of the following fields:

No.	Field	Include in API	Field name in API	Field data type in API
1	header,number	Y	ChangeNumber	StringType
2	qcintegration.id	Y	QCEntityID	IntType
3	sysmodtime	Y	Modified	DateTimeType

▼ **Web Services API properties**

Field rendering in service oriented interface.

Include in API ☒

Field name in API:

Field data type in API:



**Note:** The caption value must be unique and alphanumeric (no spaces) with the first letter capitalized (AValidCaption123, AnotherValidCaption, and so on).

- Click **Toolkit > WSDL Configuration** and search for the `cm3r` table. Update the External Access Definition as follows based on the `cm3r` table.

No.	Field	Value
1	Service Name	QCIntChangeService
2	Object Name	QCIntChange
3	Allowed Actions	save
4	Action Names	Update

**External Access Definition**

Service Name:

Name:   

Object Name:

Allowed Actions Expressions Data Policy

Allowed Actions	Action Names
save	Update

## 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.  <b>Note:</b> No spaces between slashes.

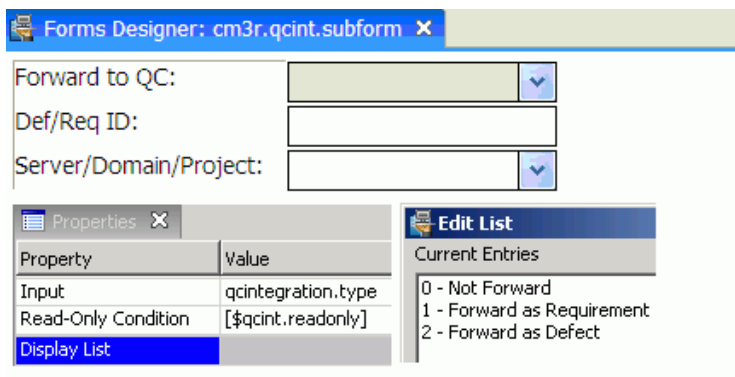
Save this global list and click **Rebuild Global List** in the Options menu.

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	<b>Caption:</b> Forward to QC:

Component	Properties
Combo Box	<ul style="list-style-type: none"> <li>■ <b>Input:</b> qcintegration.type</li> <li>■ <b>Value List:</b> 0;1;2</li> <li>■ <b>Display List:</b> 0 - Not Forward;1 - Forward as Requirement;2 - Forward as Defect</li> <li>■ <b>Select Only:</b> Yes</li> <li>■ <b>Read-Only Condition:</b> [\$qcint.type.readonly]</li> </ul>
Label	<b>Caption:</b> Def/Req ID:
Text	<ul style="list-style-type: none"> <li>■ <b>Input:</b> qcintegration.id</li> <li>■ <b>Read-Only:</b> Yes</li> </ul>
Label	<b>Caption:</b> Server/Domain/Project:
Combo Box	<ul style="list-style-type: none"> <li>■ <b>Input:</b> qcintegration.project</li> <li>■ <b>Value List:</b> \$G.qcintegration.change.project</li> <li>■ <b>Read-Only Condition:</b> [\$qcint.project.readonly]</li> <li>■ <b>Mandatory Condition:</b> [qcintegration.type]&gt;0</li> </ul>



## 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:

The screenshot shows the 'Forms Designer' window for a 'Request For Change' form. The main area displays a form with fields for RFC No., Phase, Status, Approval Status, Alert Stage, Reason for Change, Planned Start, Planned End, Risk Assessment, Initial Impact Assessment, Urgency, Priority, and Folder. A tab labeled 'qc integration' is selected. To the right, the 'Properties' panel shows the 'Subformat' properties for the selected subform.

Property	Value
Name	subview1208421811523
X	14
Y	2
Width	90
Height	16
Visible	<input checked="" type="checkbox"/>
Visible Condition	
Format	cm3r.qcint.subform
Virtual Join	<input type="checkbox"/>
Display Blank	<input checked="" type="checkbox"/>
Display Using Table	<input type="checkbox"/>
Input	

4. Save the changes.

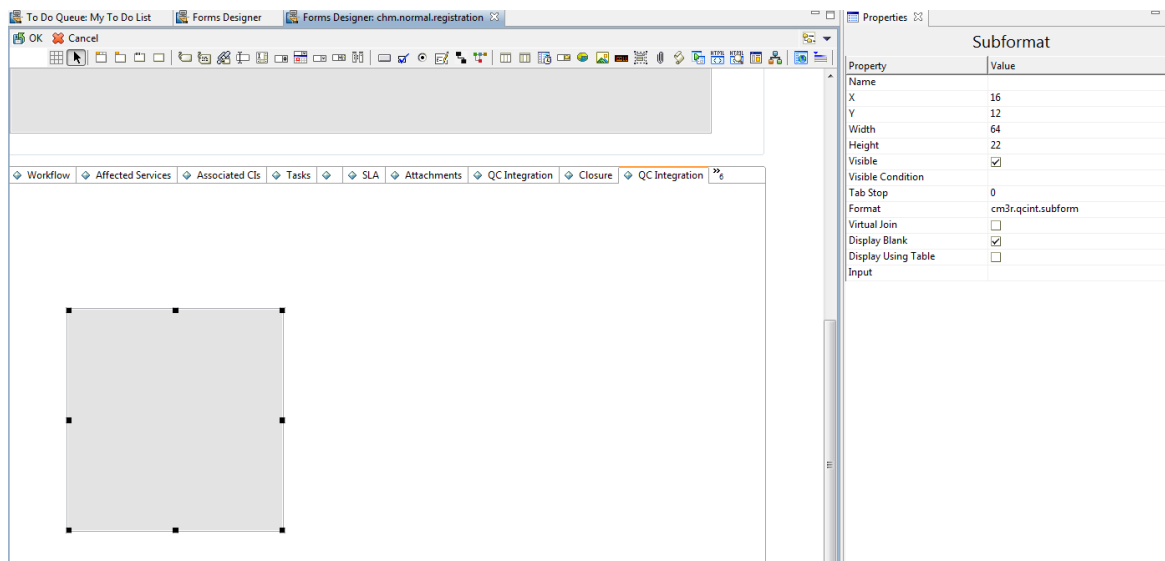
**Note:** If the error message "Format 'cm3r.qcint.subform' not found (display, show.rio)" appears, log out and then log in again to enable the subform.

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.



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:



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 59](#) for configurations in the PD environment and the SM 9.4x Codeless environment.

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

1. 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	\$qcint.type.readonly=2;if (qcintegration.type in \$file~=0) then (\$qcint.type.readonly=1)
true	true	\$qcint.project.readonly=2;if (qcintegration.type in \$file~=0 and not null (qcintegration.project in \$file)) then (\$qcint.project.readonly=1)

**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    Queries    **Calculations**    JavaScript    Validations    Subroutines    Addl Options    Privileges

**Format Control Maintenance - Calculations**

Name: cm3r.rfc.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	\$qcint.readonly=2;if (qcintegration.type in \$file="1" or qcintegration.type in \$file="2") then (\$qcint.readonly=1)

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.

No	Parameter	Value
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	<input type="text" value="not null(qcintegration.project in \$file)"/>	Delete	<input type="text"/>
Message	<input type="text" value="The Server/Domain/Project is required."/>	Display	<input type="text"/>
Comments	<input type="text"/>	Initial	<input type="text"/>
Add	<input type="text" value="qcintegration.type in \$file~=0"/>	Set Focus to	<input type="text" value="qcintegration.project"/>
Update	<input type="text" value="qcintegration.type in \$file~=0"/>	Message ID	<input type="text"/>

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. 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:

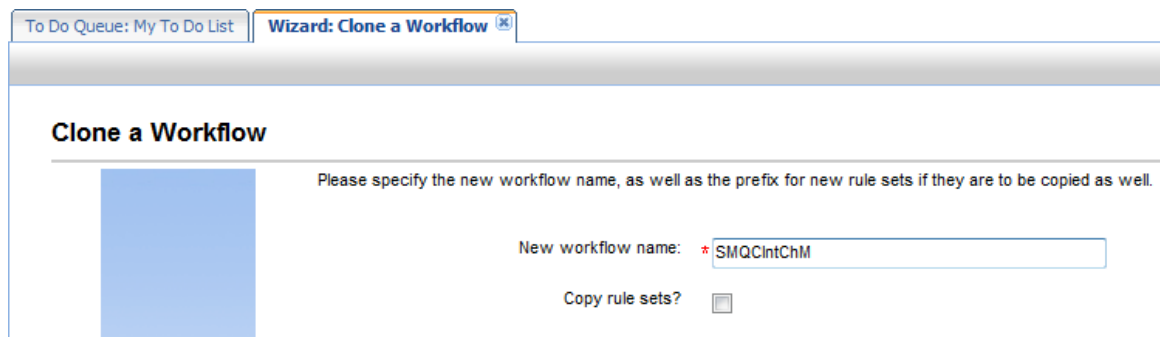
1. ["Copy an Existing Workflow" below](#)
2. ["Associate an Existing Change Category with the New Workflow" on the next page](#)
3. ["Create New Rule Set for Initialization and Validation" on page 61](#)
4. ["Associate the New Workflow with the New Rule Set" on page 64](#)

## 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.



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.

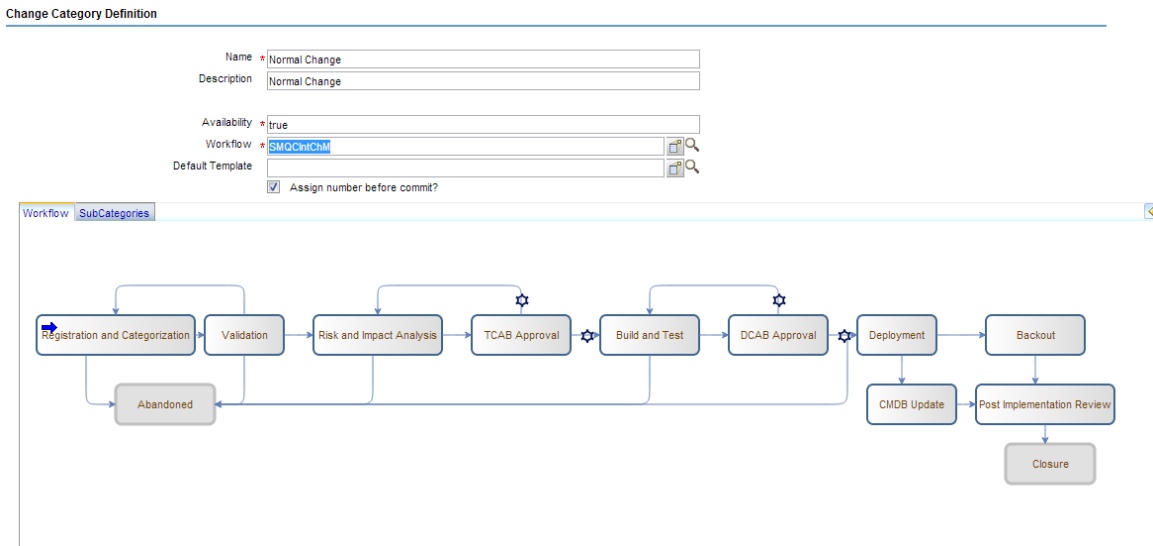
## Associate an Existing Change Category with the New Workflow

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.

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



## Create New Rule Set for Initialization and Validation

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

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

- Click **New** and **Save**.
- Click **Add Rule**.
- 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 Change Record
Statement	<pre> vars['\$qcint.type.readonly'] = 2;  vars['\$qcint.project.readonly'] = 2;  var _null=system.functions._null;  var file = vars.\$L_file;  if(file["qcintegration.type"] !=0 &amp;&amp; !_null(file["qcintegration.type"])) {  vars['\$qcint.type.readonly'] = 1  }  if(file["qcintegration.type"] !=0 &amp;&amp; !_null(file["qcintegration.project"])) {  vars['\$qcint.project.readonly'] = 1  } </pre>

7. Click **OK**.
8. Click **Save** and **Exit**.

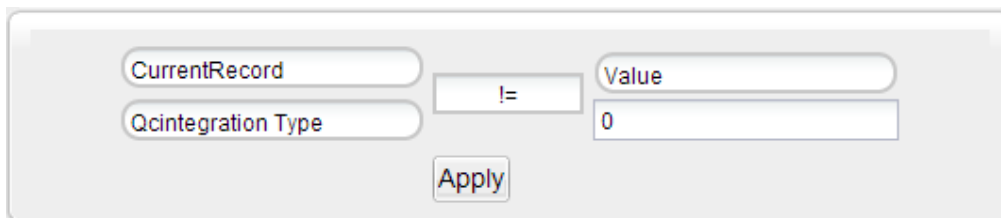
To create a new rule set for validation, follow the steps below:

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

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

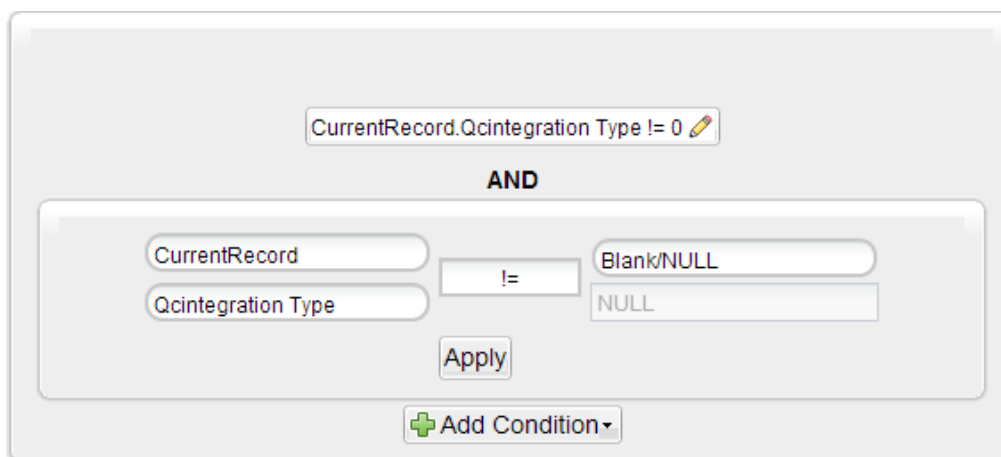
Field	Value
Table name	cm3r
HP Proprietary	

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 an expression as illustrated in the following screenshot:



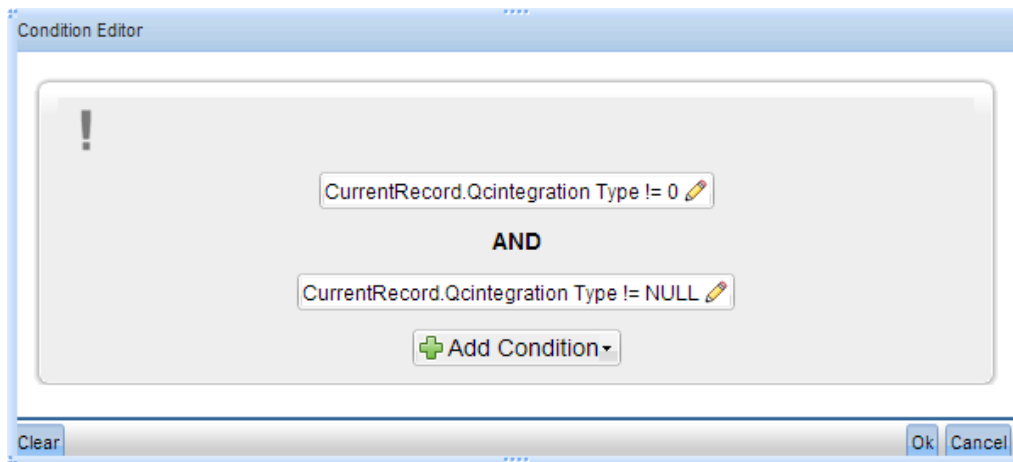
The screenshot shows a rule editor interface. It has two input fields on the left: 'CurrentRecord' and 'Qcintegration Type'. In the center is a dropdown menu showing '!='. To the right of the dropdown is another input field labeled 'Value' with '0' entered. Below these elements is an 'Apply' button.

8. Click **Apply**.
9. Add condition, Select the AND operator, and then add another expression as illustrated in the following screenshot:



The screenshot shows a rule editor interface with two conditions. The first condition is 'CurrentRecord.Qcintegration Type != 0' with a pencil icon. Below it is the 'AND' operator. The second condition is shown in a separate box with input fields for 'CurrentRecord' and 'Qcintegration Type', a dropdown for 'Value' with 'Blank/NULL' selected, and an 'Apply' button. At the bottom of the interface is a '+ Add Condition' button.

10. Click **Apply**.



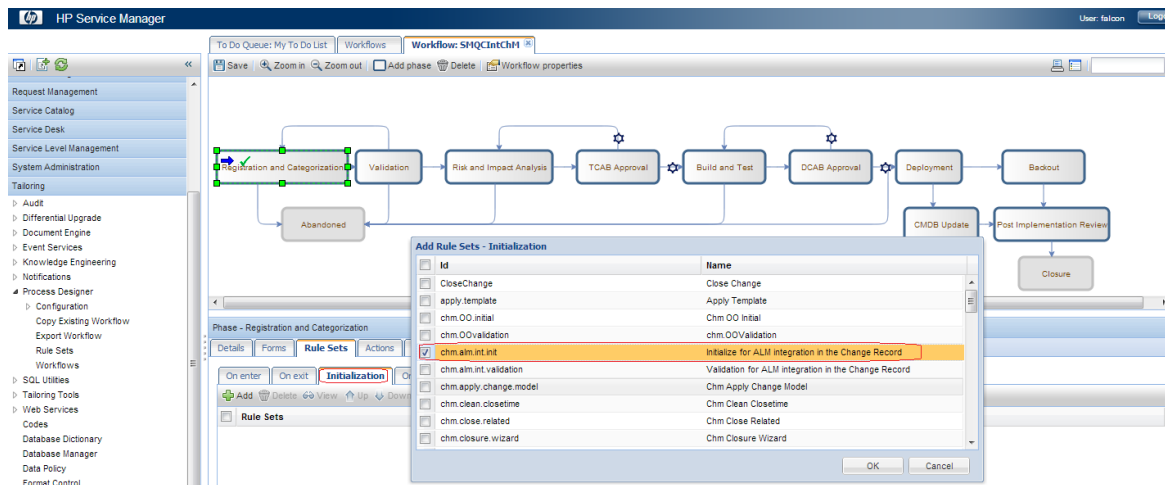
11. Click **OK** to exit the Condition Editor.
12. Click **OK** to exit the Set Mandatory Fields page.
13. Click **Save** and **Exit**.

## Associate the New Workflow with the New Rule Set

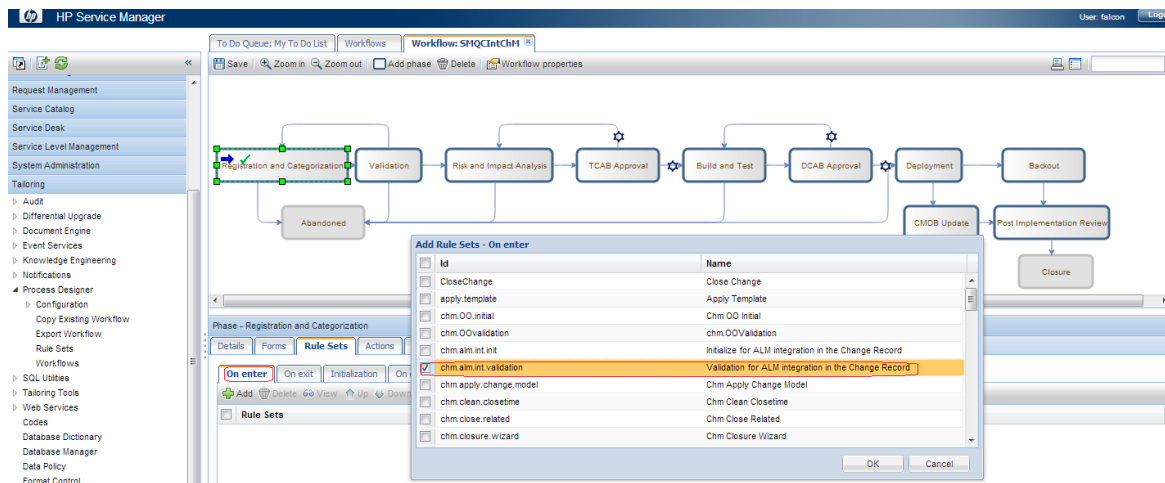
To associate the new workflow with the new rule set, follow the steps below:

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.





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.



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 72](#)

### On QC 10 or 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 69](#)
4. ["Verify" on page 70](#)

### 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
BG_USER_XX	Change ID	String
BG_USER_XY	Created from	String

The following figure shows an example project entity.

Quality Center - Project Customization

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

Cycle

Defect

System Fields

User Fields

Change ID

Created from

Release

Release Folder

Requirement

Run

Test

Test Instance

Test Set

Test Step

Field Settings

Field Name:

BG\_USER\_02

Field Label:

Change ID

Field Type:

String

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**.
2. Select **Defects module script**.



3. 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**.



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
```

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.

## Verify

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

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

**New Defect**

Clear Attach: [Icons]

\* Summary: [Text Field]

Details SM Integration (New)

Change ID: [Text Field] Created from: [Text Field]

Submit Close

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 Details**

Defect: 13 a new defect

Details SM Integration (Details)

Change ID: C18 Created from: Created from SM/SC

Details Attachments 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](#)

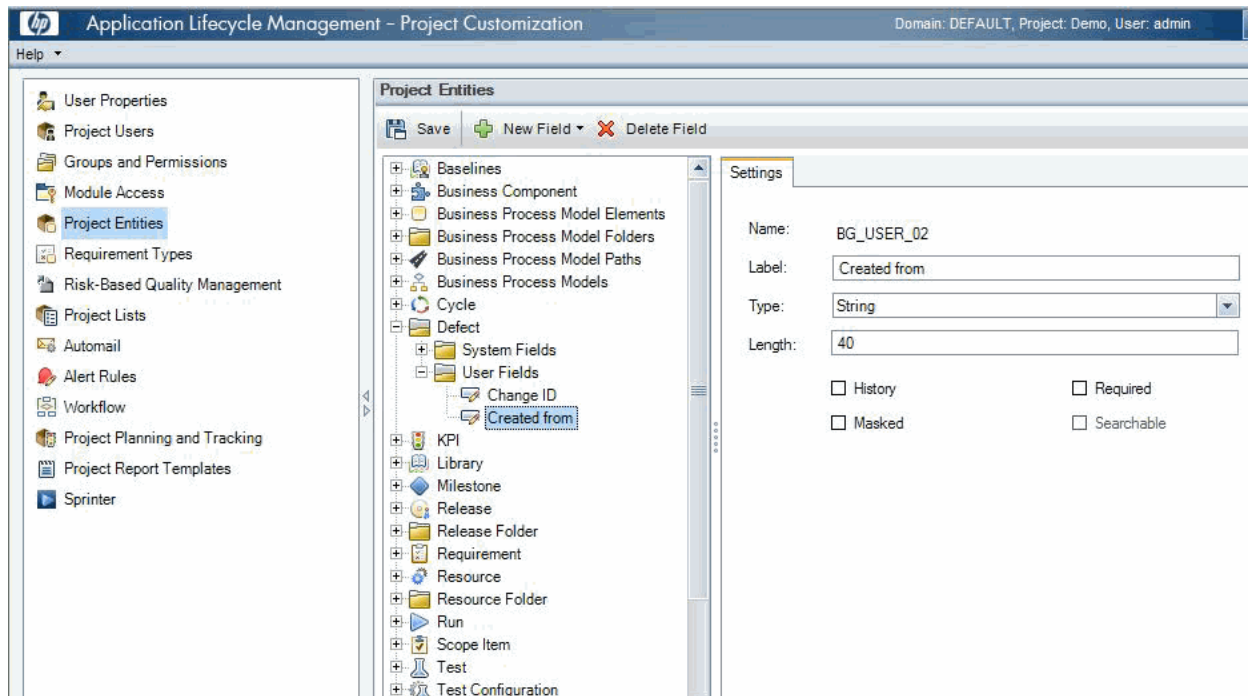
## Add fields

To add the required fields for Defect module customization:

1. Log on to ALM as a project administrator.
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
BG_USER_XX	Change ID	String
BG_USER_XY	Created from	String





4. Click **Workflow > Script Editor**.
5. Select **Defects module script**.
6. Set the **Readonly** fields by adding the following lines to the **Bug\_New** and **Bug\_Move** to 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:

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

The screenshot shows the 'New Defect' form in HP Service Manager. The form is titled 'New Defect' and has a blue header bar. Below the header is a toolbar with icons for back, forward, and other actions. The main area is divided into a left sidebar and a right main panel. The sidebar has 'Details' and 'Attachments' sections. The main panel has a 'Summary' field at the top, followed by a 'Details' section with various fields: 'Reproducible' (Y), 'Assigned To' (dropdown), '\* Severity' (dropdown), 'Status' (New), 'Subject' (dropdown), 'Target Cycle' (dropdown), 'Target Release' (dropdown), 'Change ID' (read-only, circled), 'Created from' (read-only, circled), 'Problem ID' (dropdown), and 'Modified' (dropdown). Below these is a 'Description' field. At the bottom are 'Submit', 'Close', and 'Help' buttons.

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

Defect Details

Defect ID: 11 \* Summary: Upgrade internal memory from 512 MB to 2048 MB

Details

Attachments  
Linked Entities  
History

Reproducible: [ ] Assigned To: [ ]

\* Severity: 2-Medium Status: New

Subject: [ ] Target Cycle: [ ]

Target Release: [ ] Change ID: C10003

Created from: [ ] Problem ID: [ ]

Modified: 2010-11-4 13:56:50

Description: [ ] Comments: [ ] Add 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 80](#)
4. ["Test the Link" on page 80](#)

### 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>/qcbn.`

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			
Type	QC Field	Direction	SM ChangeManagement Field
	Severity	<---->	Urgency
	Change ID	<----	ChangeNumber
	Defect ID	---->	QCEntityID
	Summary	<---->	Description
	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 → QC/ALM Defect 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 <sup>2</sup>	String	Change ID	String	<-	String	ChangeNumber	StringType	header,number
10	BG_BUG_ID	Number	Defect ID <sup>3</sup>	Number	->	Number	QCEntityID <sup>4</sup>	IntType	qcintegration.id
255	BG_SUMMARY	String	Summary	String	<->	String	Description	StringType	description.struct description
70	BG_SEVERITY	Lookup List	Severity	Single value list	<->	Single value list	Urgency	Character	severity

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

QC Defect Schema		SM ChangeManagement	
Name	Type	Name	Type
Severity	Single value list	Urgency	Single value list

Mapped Fields			
Type	QC Field	Direction	SM ChangeManagement Field
	Change ID	<---	ChangeNumber
	Defect ID	---	QCEntityID
	Summary	<---	Description
	Severity	<---	Urgency

Mapping properties			Value mapping	Field properties
Endpoint 1 value	Direction	Endpoint 2 value		
5-Urgent	<---	1 - Critical		
4-Very High	<---	2 - High		
3-High	<---	3 - Average		
2-Medium	<---	4 - Low		

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

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

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

1

Lookup list is created in QC.

The screenshot shows the 'Field Settings' dialog box. Under 'Field Settings', 'Field Name' is 'BG\_SEVERITY', 'Field Label' is 'Severity', and 'Field Type' is 'Lookup List'. Below this, the 'Lookup List' section has a dropdown menu showing 'Severity', and buttons for 'New List' and 'Goto List'. A 'Verify Value' checkbox is checked. To the right, the 'Project Lists' pane shows a list of items: 1-Low, 2-Medium, 3-High, 4-Very High, and 5-Urgent.

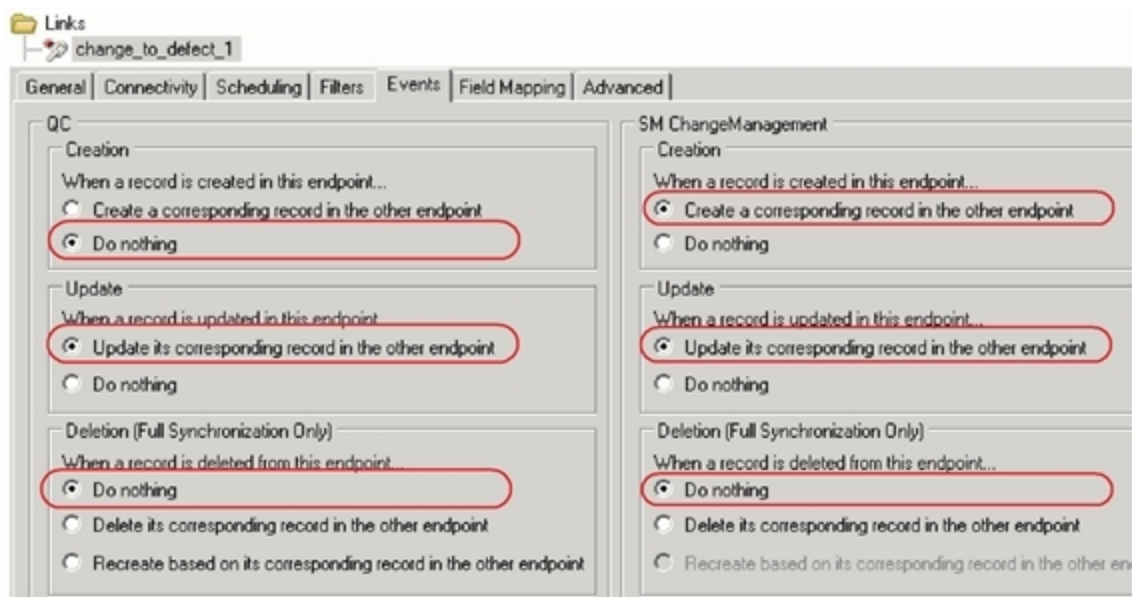
<sup>2</sup>The XML file is in <QCS\_Install\_Dir>\adapters\dat\SM ChangeManagement\configuration\_file\_default.xml (see ["SM Change Management Example" on page 21](#)).

## 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:



## 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).

Category	Description
RFC	Request For Change

**Request For Change**

General Description

Description of Change

Change to Defect

4. Change the phase to **Building**. The QC Integration tab appears.

Phase Name	Description
Assessment	Assessment
<b>Building</b>	<b>RFC Building</b>
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

Forward to QC: 2 - Forward as Defect

Def/Req ID:

Server/Domain/Project: localhost/DEFAULT/Demo

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.

The screenshot shows the 'Defect Details' window. At the top, there's a toolbar with navigation icons. Below it, the 'Defect:' field contains '9' and the description field contains 'change to defect'. The 'Details' tab is selected, showing 'Change ID: C19' and 'Created from: Created from SM/SC'. The left sidebar has icons for 'Details', 'Attachments', 'Linked Entities', and 'History'. At the bottom, there's an 'Execution Report' link and 'OK' and 'Cancel' buttons.

**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/ServiceCenter for Change Management" below](#)
- ["Customizing the QC/ALM Requirements Module" below](#)
- ["Configuring Links in QC/ALM Synchronizer" on page 90](#)

### Customizing Service Manager/ServiceCenter for Change Management

For details, see the ["Customizing Service Manager/ServiceCenter 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 88](#)

#### 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 85](#)

3. ["Add Fields to Tabs" on page 87](#)
4. ["Create the SM Incoming Changes Folder" on page 89](#)

## 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_XX	Change ID	String
RQ_USER_XY	Created from	String

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

This is shown in the following figure.



4. 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).

## 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"
```

```

        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)"

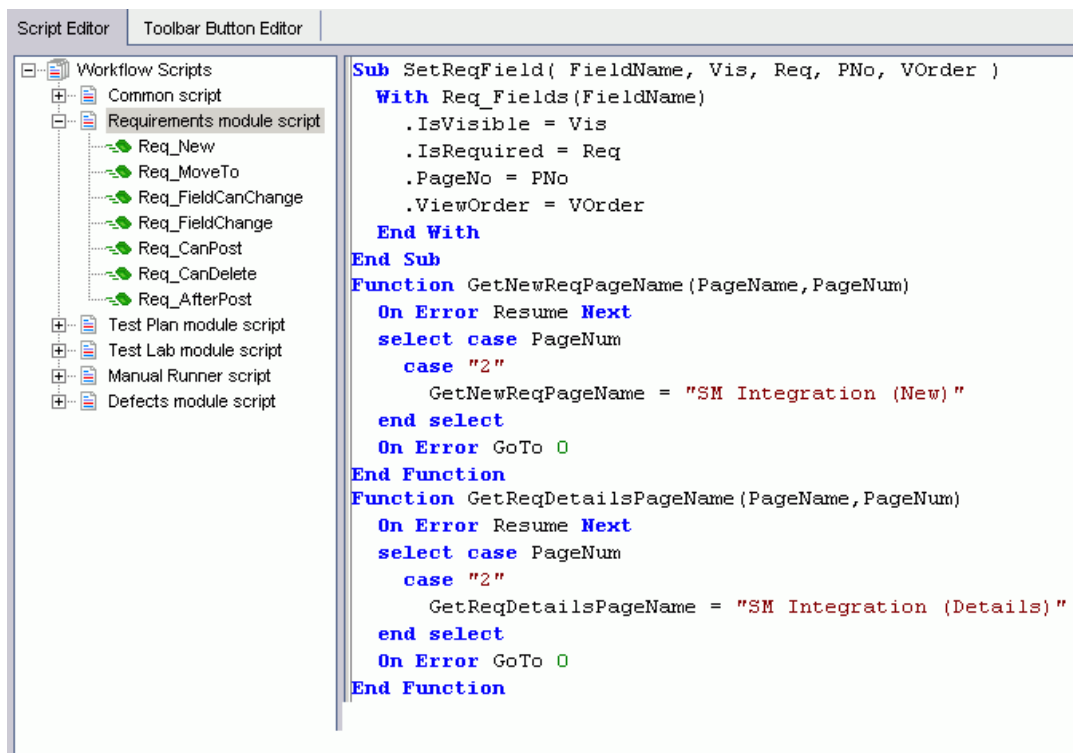
    end select

    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).

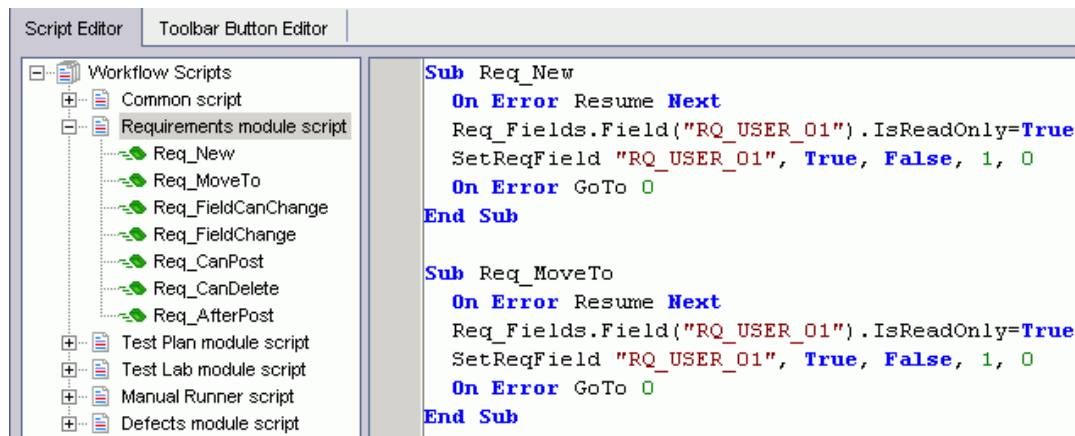
```
Req_Fields.Field("RQ_USER_XX").IsReadOnly=True
```

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

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

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

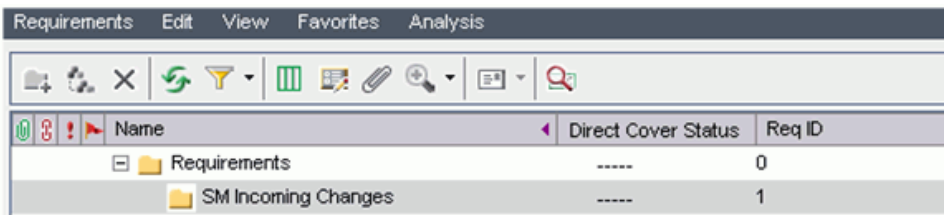
The resulting script is shown in the following screenshot:



## 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.



## 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.

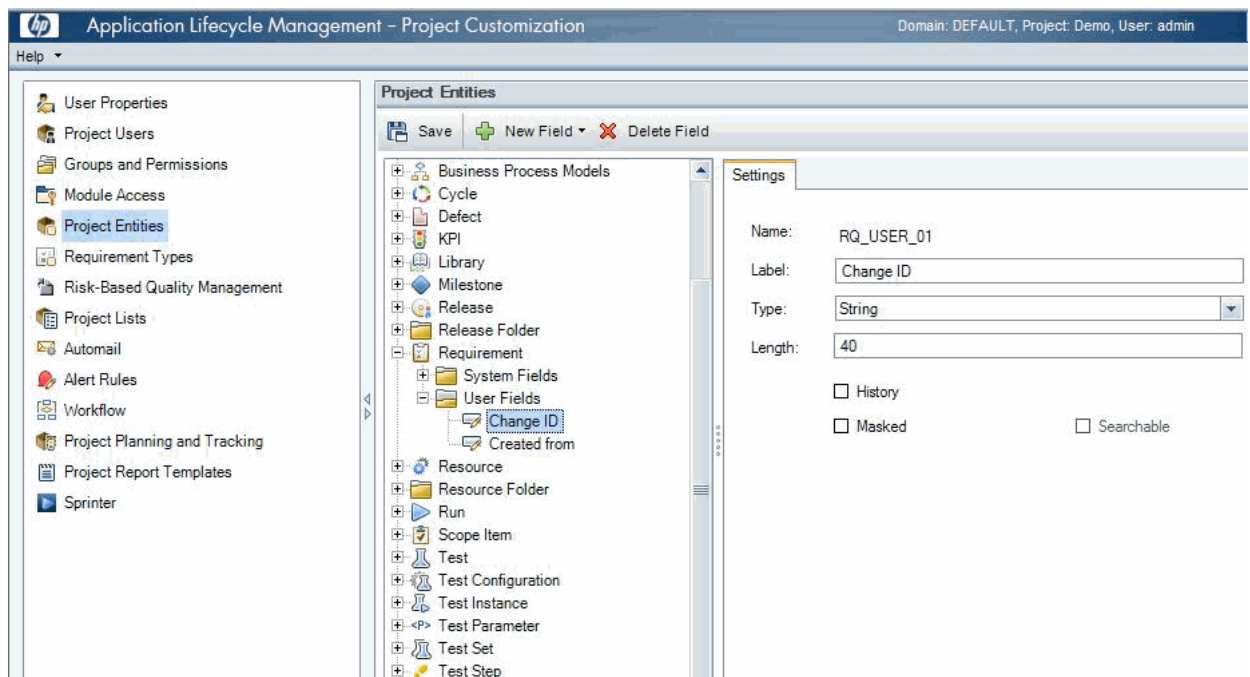
1. Log on to ALM as a project administrator.
2. Click **Tools / Customize**. The "Application Lifecycle Management - Project Customization" module opens.
3. Add the following fields for the requirement entity in project entities (XX and XY are sequential numbers auto-generated by ALM).

Field Name	Field Label	Field Type
RQ_USER_XX	Change ID	String
RQ_USER_XY	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.





4. 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 92](#)
- ["Define Events" on page 94](#)
- ["Test the Link" on page 94](#)

### 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:

QC

User name: SMQCIntUser

Password: .....

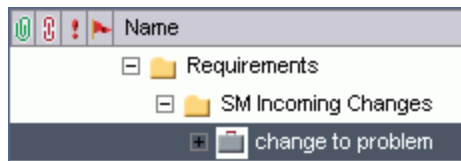
Parameter	Value
Domain	DEFAULT
Project	Demo
ServerURL	http://localhost:8080/qcbin

Check Connectivity

☒ Use alternate root folder:

Requirements\ <your\_specified\_incoming\_requirement\_folder>

Requirements will be created in the specified folder in QC.



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

General Connectivity Scheduling Filters Sub Types Mapping

Endpoint 1 Available Sub Types: Functional, Testing, Undefined

Endpoint 2 Available Sub Types:

Add Mapping >>

Mapped Sub Types: Business - SM Change

Configure

Requirement Details

Req ID: 6

\* Name: change to problem

\* Requirement Type: Business

Business

Folder

Functional

Group

Testing





Undefined

## 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:

Mapped Fields			
Type	QC Field	Direction	SM ChangeManagement Field
	Name	<---->	Description
	Change ID	<---->	ChangeNumber
	Req ID	---->	QCEntityID
	Created from	<---->	<i>Value: Created from SM/SC</i>

## 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 -> QC/ALM Requirement Mappings

QC Len	QC DB Name	QC Type	QC/QC S Label	QC/ QC S Type	Dir	QC S SM Type	QC S 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

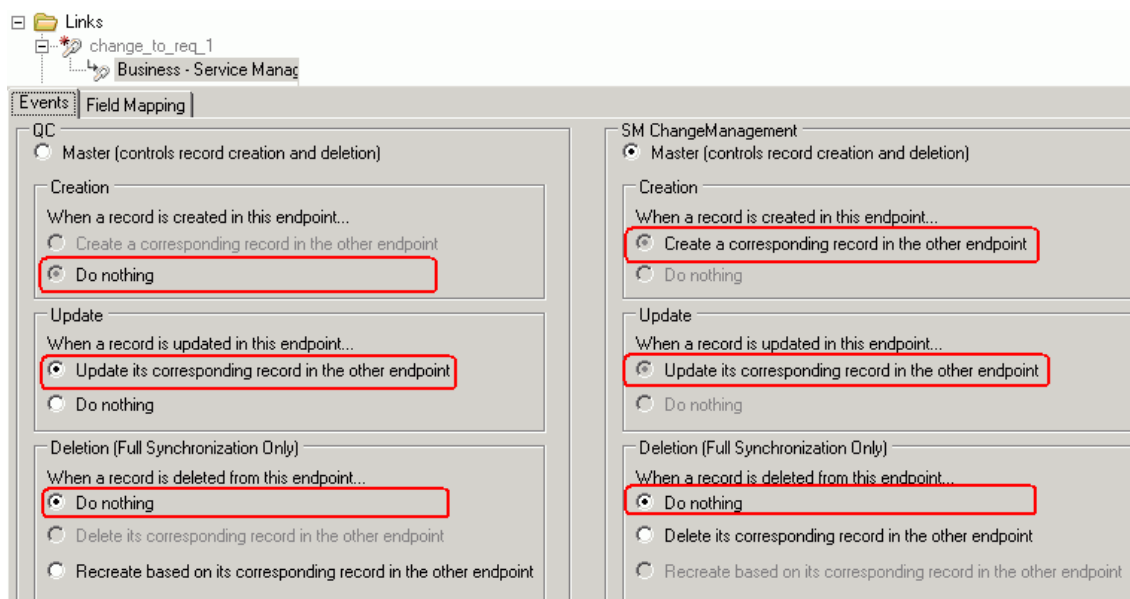
**Note:** Check for QCEntityID mapping property **Synchronize back on create**.

## 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:

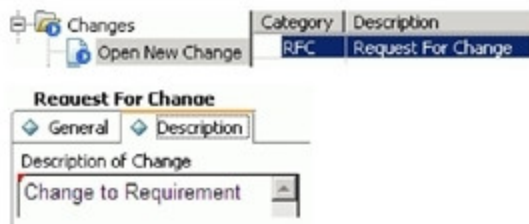


## Test the Link

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.

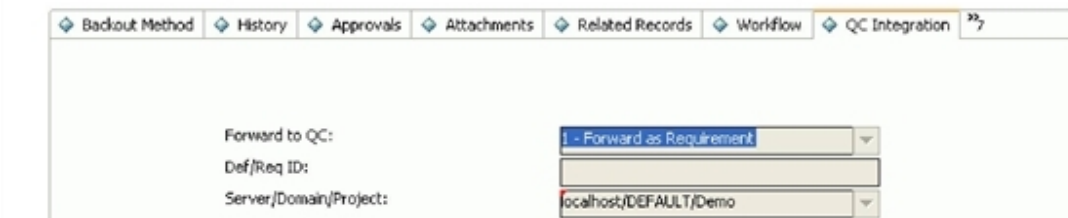
1. Save the configuration (an integrity check is automatically run).
2. Click **Enable Link**.
3. Create a Service Manager change (the category of the change depends on each Service Manager customization; RFC is used in this example).



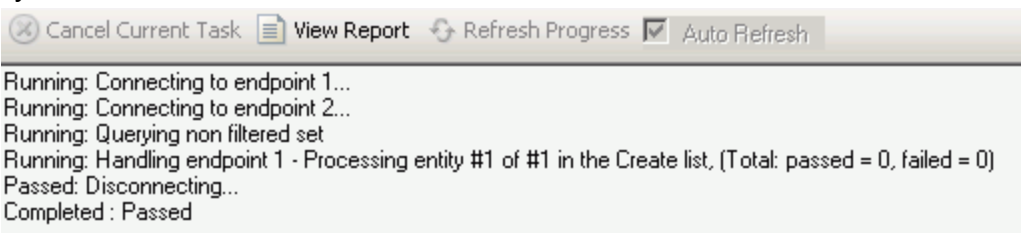
4. Change the phase to **Building**. The “QC Integration” tab appears.



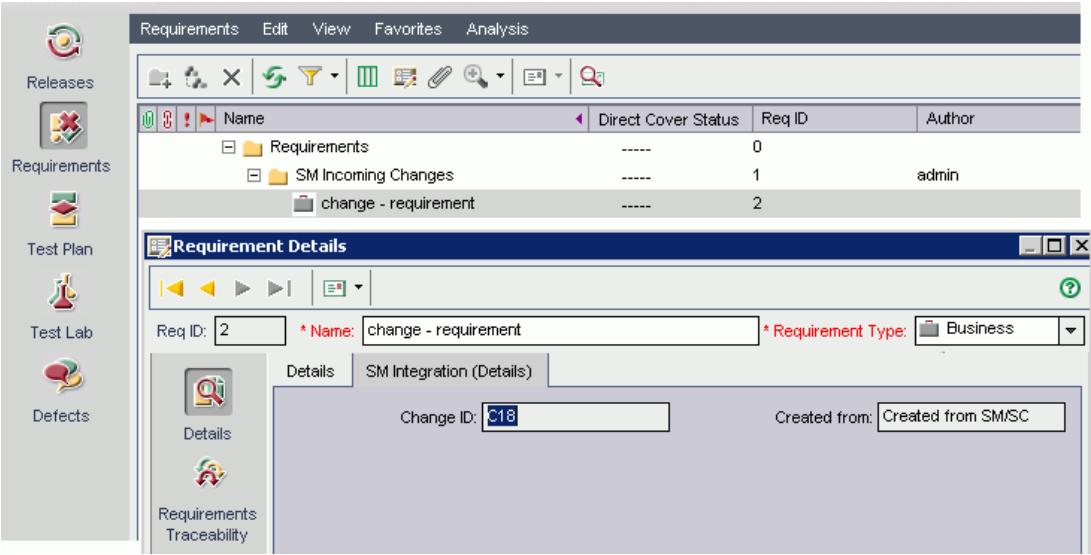
5. Select **Forward as Requirement**.



6. Synchronize.



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

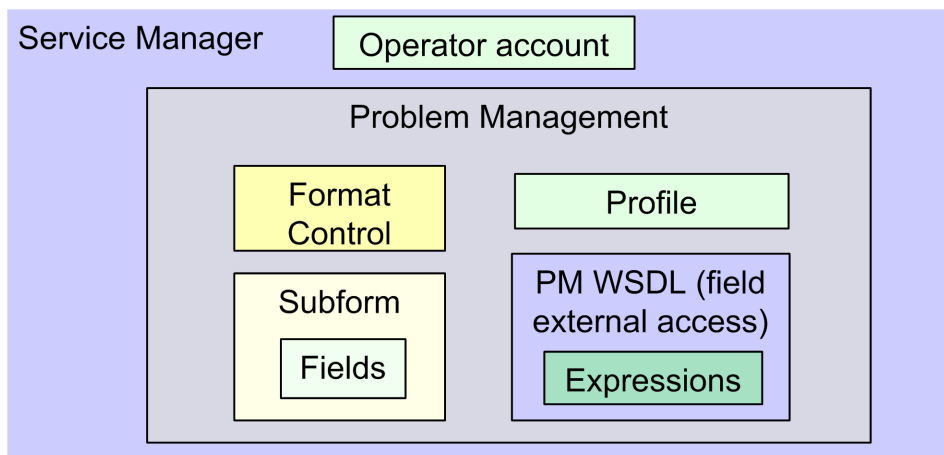
This chapter describes how to synchronize SM Problems with QC/ALM Defects ("Problem -> Defect").

This chapter includes:

- ["Customizing Service Manager/ServiceCenter for Problem Management" below](#)
- ["Customizing the QC/ALM Defects Module" on page 116](#)
- ["Configuring Links in QC/ALM Synchronizer" on page 122](#)

### Customizing Service Manager/ServiceCenter for Problem Management

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



To customize Service Manager/ServiceCenter 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](#) or ["Specify the External Access Definition on ServiceCenter" on page 103](#)
3. ["Create a Subform" on page 105](#)

4. ["Add the Subform to a Form" on page 108](#)
5. ["Add Format Control Calculations/Validations" on page 109](#)

## Add Fields

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

Field	Type	
	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.

1. Create a custom External Access Definition `QCIntProblemService` by clicking **Tailoring > WSDL configuration** on Service Manager 7.0x; or clicking **Tailoring > Web Services > WSDL 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:

Name:  ▼ Object Name:

Allowed Actions
Expressions
Fields

Allowed Actions	Action Names	Action Type
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	Type
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 Definition

Service Name: QCIntProblemService

Name: rootcause

Allowed Actions

Expressions

Fields

Field	Caption	Type
qcintegration.id	QCEntityID	IntType
id	ProblemID	StringType
sysmodtime	Modified	DateTimeType
qcintegration.project	QCProject	StringType
incident.category	Category	StringType
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 caption value must be unique and alphanumeric (no spaces) with the first letter capitalized (AValidCaption123, AnotherValidCaption, and so on). The above values are required (Do NOT change 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.
- 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)

No.	Expression
2	<code>\$rc.update=update in \$L.file;if (dennull(\$rc.update)={}) then (\$rc.update={"QC update sent"})</code>
3	<code>if (\$L.need.to.update=true) then (\$rc.update={"QC update sent"})</code>
4	<code>update in \$L.file=update in \$L.file.save</code>

**External Access Definition**

Service Name:  Name:  Object Name:

Allowed Actions Expressions Fields

**Expressions**

```
cleanup($pm.activity);cleanup($rc.update);if same(update in $L.file, update in $L.file.save) then ($L.need.to.update=true)
$rc.update=update in $L.file;if (dennull($rc.update)={}) then ($rc.update={"QC update sent"})
if ($L.need.to.update=true) then ($rc.update={"QC update sent"})
update in $L.file=update in $L.file.save
current.phase in $L.file="Problem 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.

1. Refer to [step 1](#).
2. Enable the required fields in the web service.

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

### External Access Definition

Service Name: \* QCIntProblemService ☐ Released:  
 Name: \* rootcause ☐ Deprecated:  
 Object Name: QCIntProblem

Allowed Actions Expressions **Fields**

Field	Caption	Type
id	ProblemID	StringType
sysmodtime	Modified	DateTimeType
qcintegration.id	QCEntityID	IntType
qcintegration.project	QCProject	StringType
qcintegration.type	QCIntegrationType	StringType
qcintegration.created.from	CreatedFrom	StringType

**Note:** The caption value must be unique and alphanumeric (no spaces) with the first letter capitalized (AValidCaption123, AnotherValidCaption, and so on). The above values are required (Do NOT change 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	\$rc.update=update in \$L.file;if (denull(\$rc.update)={}) then (\$rc.update={"QC update sent"})
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 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

```
cleanup($pm.activity);cleanup($rc.update);if same(update in $L.file, update in $L.file.save) then ($L.need.to.update=true)
$rc.update=update in $L.file;if (dnull($rc.update)={}) then ($rc.update={"QC update sent"})
if ($L.need.to.update=true) then ($rc.update={"QC update sent"})
update in $L.file=update in $L.file.save
isKnownError in $L.file=nullsub(isKnownError in $L.file,true)
```

## Specify the External Access Definition on ServiceCenter

To specify the External Access Definition on ServiceCenter:

1. Click **System Definition > Tables > rootcause > Fields and keys definitions for the rootcause table** and change the settings of these fields.

No.	Field	Include in API	Field name in API	Field data type in API
1	id	Y	ProblemID	StringType
2	sysmodtime	Y	Modified	DateTimeType
3	qcintegration.id	Y	QCEntityID	IntType

**Note:** The caption value must be unique and alphanumeric (no spaces) with the first letter capitalized (AValidCaption123, AnotherValidCaption, and so on).

2. Click **Menu navigation > Utilities > Tools > Document Engine > Process**. Type `rca.save` in the Process Name field, and then click **Search**.
3. Change the name to `rca.qcupdate` and click **Add**.

4. Append the following lines to the Initial Expressions tab.

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

**Process Definition**

Process Name:

☐ Save Cursor Position? ☐ Run Standard Process when complete?

☐ Run in Window? Window Title:

Initial Expressions Initial Javascript RAD Final Expressions Final Javascript Next Process

```

if same(nullsub(full.name in $G.rc.environment, full.name in $G.rc.global.environment), true) then ($L.operator=nullsub($lo.ufname, nullsu...
$L.stamp=str(tod()))+" ("+$L.operator+");"
if ($rc.update={} or $rc.update={""}) then ($rc.update=NULL);if ($kne.update={} or $kne.update={""}) then ($kne.update=NULL);if ($p...
if (update in $L.file=NULL) then (update in $L.file={""})
if ($G.bg and not null($G.bg.activity.type)) then ($rc.update=nullsub($rc.update, $G.bg.activity.text);$rc.update=nullsub($rc.update, "E...
if (filename($L.file)~="rootcausetask") then if (status in $L.file="Past Due" and expected.resolution.time in $L.file>tod()) then (status in $L...

$L.save.status=status in $L.file
if (status in $L.file="Open" or status in $L.file="Reopened" and $reopen.flag=false) then (status in $L.file="Updated")
$reopen.flag=false

cleanup($pm.activity);cleanup($rc.update);if same(update in $L.file, update in $L.file.save) then ($L.need.to.update=true)
$rc.update=update in $L.file;if (denull($rc.update)={}) then ($rc.update={"QC update sent"})
if ($L.need.to.update=true) then ($rc.update={"QC update sent"})
update in $L.file=update in $L.file.save

```

5. Click **Save**.
6. Update the `rca.view` state record .
- Click **Utilities > Tools > Document Engine > States**. Type the state name `rca.view` and click **Search**.
  - Add the following row to the Non-base methods table, and then click **Save**.



Display Action	Process Name	Condition	Save First
qcupdate	rca.qcupdate	\$L.mode~="close" and \$L.mode~#"add"	false



7. Update the Problem Management integration web service.

- Click **Toolkit > WSDL Configuration**. In the Name field type `rootcause`, and then click **Search**.
- Update the External Access Definition as follows based on table `rootcause`.

No.	Field	Value
1	Service Name	QCIntProblemService
2	Object Name	QCIntProblem
3	Allowed Actions/Action Names	add/Create
4	Allowed Actions/Action Names	qcupdate/Update

**External Access Definition**

Service Name:

Name:   

Object Name:

◆ Allowed Actions ◆ Expressions ◆ Data Policy

Allowed Actions	Action Names
add	Create
qcupdate	Update

**Note:** Delete all Allowed Actions without an Action Name.

## Create a Subform

To create a subform:

- Create a global list.
  - 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	
5	User Defined List?	Yes	Check box
6	Value List	{"server1/domain1/project1", "server2/domain2/project2"}	Change to the values for your system  <b>Note:</b> 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	<b>Caption:</b> Synchronize with QC:

No.	Component	Properties
2	Combo Box	<ul style="list-style-type: none"> <li>■ <b>Input:</b> qcintegration.type</li> <li>■ <b>Value List:</b> 0;1;</li> <li>■ <b>Display List:</b> 0 - Not Synchronize;1 - Synchronize with QC Defect</li> <li>■ <b>Select Only:</b> Yes</li> <li>■ <b>Read-Only Condition:</b> [\$qcint.type.readonly]</li> </ul>
3	Label	<b>Caption:</b> Defect ID:
4	Text	<ul style="list-style-type: none"> <li>■ <b>Input:</b> qcintegration.id</li> <li>■ <b>Read-Only:</b> Yes</li> </ul>
5	Label	<b>Caption:</b> Server/Domain/Project:
6	Combo Box	<ul style="list-style-type: none"> <li>■ <b>Input:</b> qcintegration.project</li> <li>■ <b>Value List:</b> \$G.qcintegration.problem.project</li> <li>■ <b>Read-Only Condition:</b> [\$qcint.project.readonly]</li> <li>■ <b>Mandatory Condition:</b> [qcintegration.type]&gt;0</li> </ul>

OK Cancel

Synchronize with QC:

Defect ID:

Server/Domain/Project

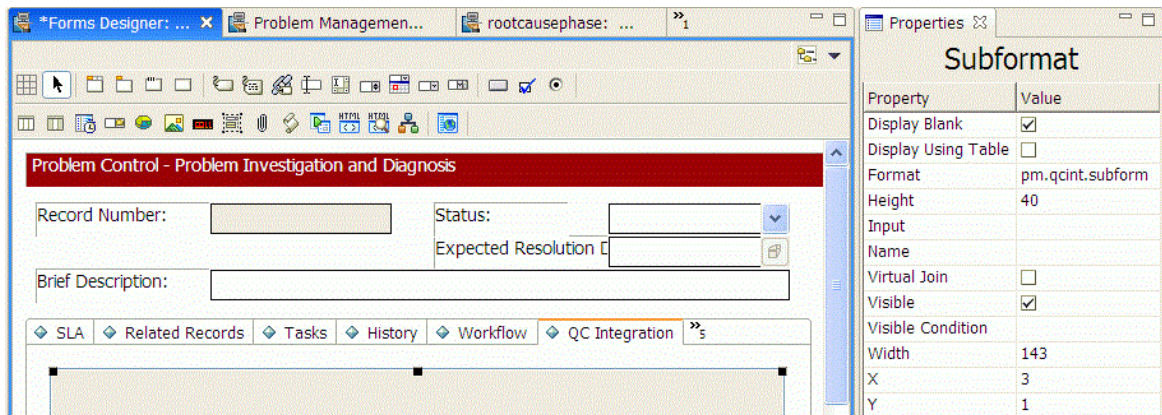
Created from:

## 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.

1. 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:



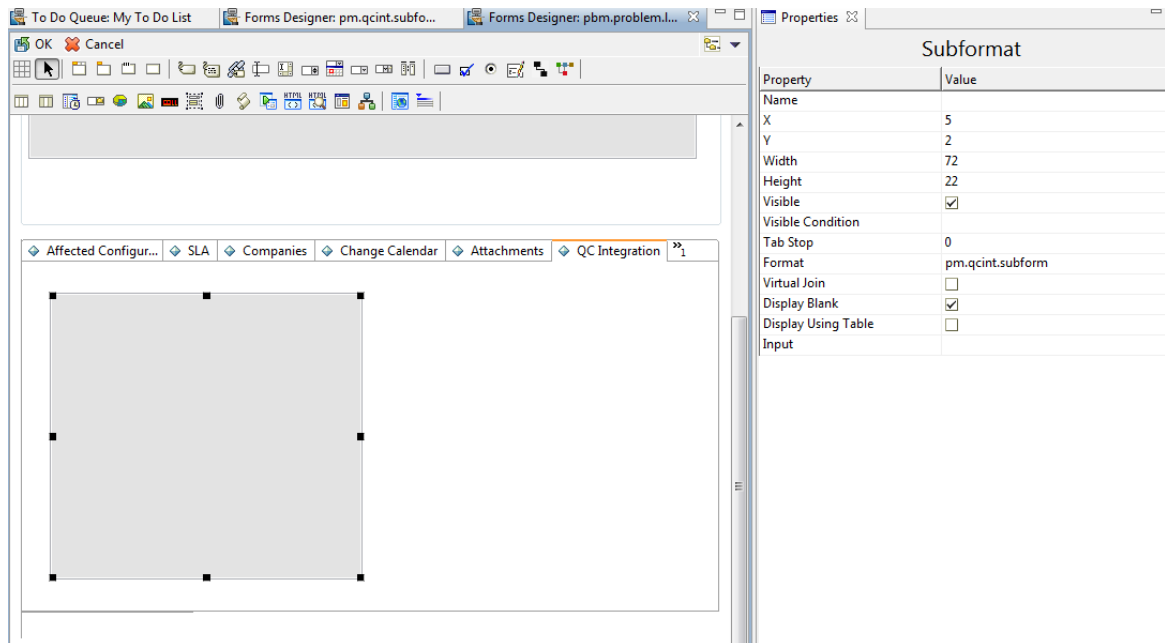
4. Save the changes.

**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.

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:



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

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 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 114](#)

## 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.

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**

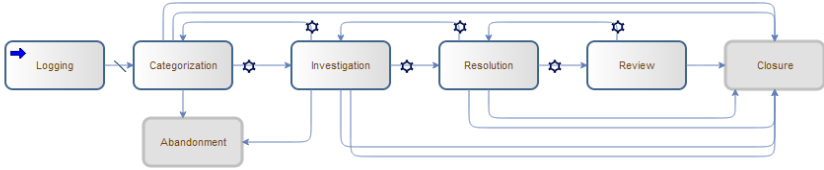
Name:  Apply To:

Active: ☒

Description:

Workflow:  

**Subcategories | Workflow**



```

graph LR
    Logging --> Categorization
    Categorization --> Investigation
    Investigation --> Resolution
    Resolution --> Review
    Review --> Closure
    Investigation --> Categorization
    Resolution --> Categorization
    Review --> Categorization
    Resolution --> Abandonment
    Review --> Abandonment
    Abandonment --> Logging
  
```

## 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.functions._null;  var file = vars.\$L_file;  if(file["qcintegration.type"] !=0 &amp;&amp; !_null(file["qcintegration.type"])) {  vars['\$qcint.type.readonly'] = 1  }  if(file["qcintegration.type"] !=0 &amp;&amp; !_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:

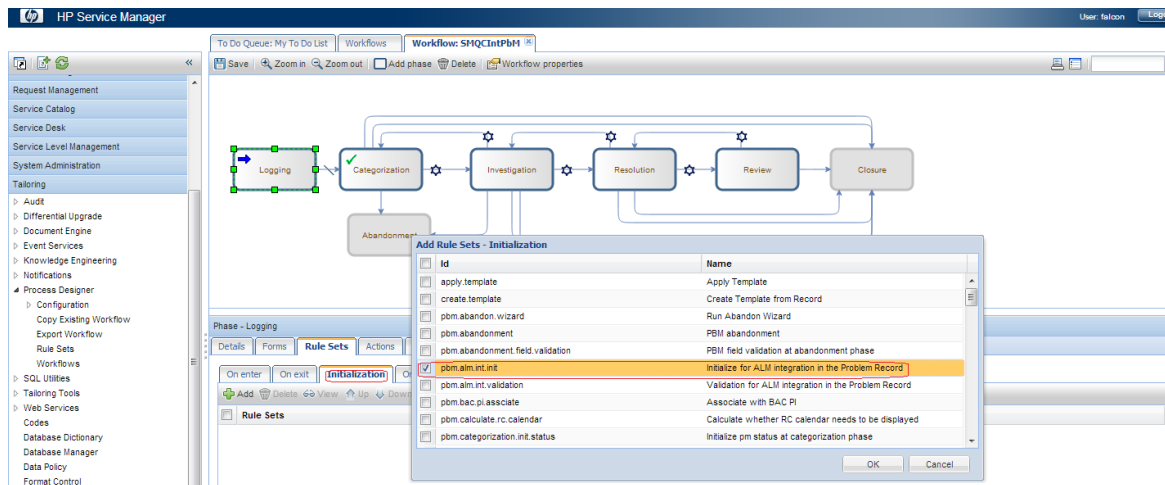
Field	Value
ID	pbm.alm.int.validation
Available as action	False
Name	Validation for ALM integration in the Problem Record
Table name	rootcause
HP Proprietary	

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 61](#).

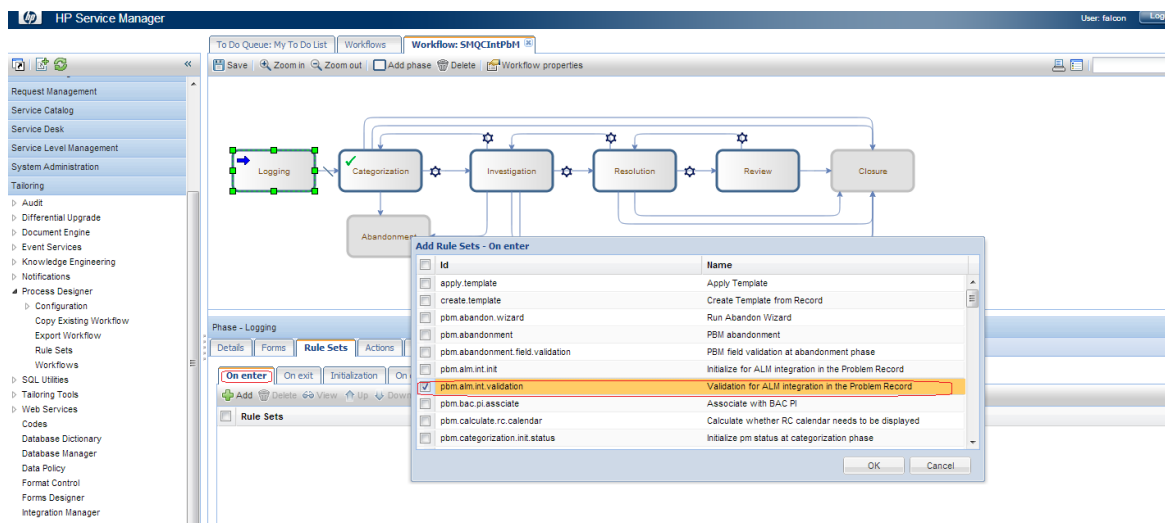
## Associate the New Workflow with the New Rule Set

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.



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.



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 120](#)

### 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 118](#)

### 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 - Project Customization

[User Properties](#)  
[Project Users](#)  
[Groups](#)  
[Module Access](#)  
**[Project Entities](#)**  
[Requirement Types](#)  
[Risk-Based Quality](#)  
[Project Lists](#)  
[Automail](#)  
[Alert Rules](#)  
[Workflow](#)

### Project Entities

Project Entities

- Defect
  - System Fields
  - User Fields
    - Change id
    - Forward as problem**
    - Problem ID

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**.
2. Select **Defects module script**.

Quality Center - Project Customization

[User Properties](#)  
[Project Users](#)  
[Groups](#)  
[Module Access](#)  
[Project Entities](#)  
[Requirement Types](#)  
[Risk-Based Quality Man](#)  
[Project Lists](#)  
[Automail](#)  
[Alert Rules](#)  
**[Workflow](#)**

### Workflow

[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.

[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 Button Editor

- Workflow Scripts
  - Common script
  - Requirements module script
  - Test Plan module script
  - Test Lab module script
  - Manual Runner script
  - Defects module script**

- GetNewBugPageName
- GetDetailsPageName**
- Bug\_New
- Bug\_MoveTo
- Bug\_FieldCanChange
- Bug\_FieldChange
- Bug\_CanPost
- Bug\_CanDelete
- Bug\_AfterPost
- SetFieldApp

3. 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**.



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_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_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

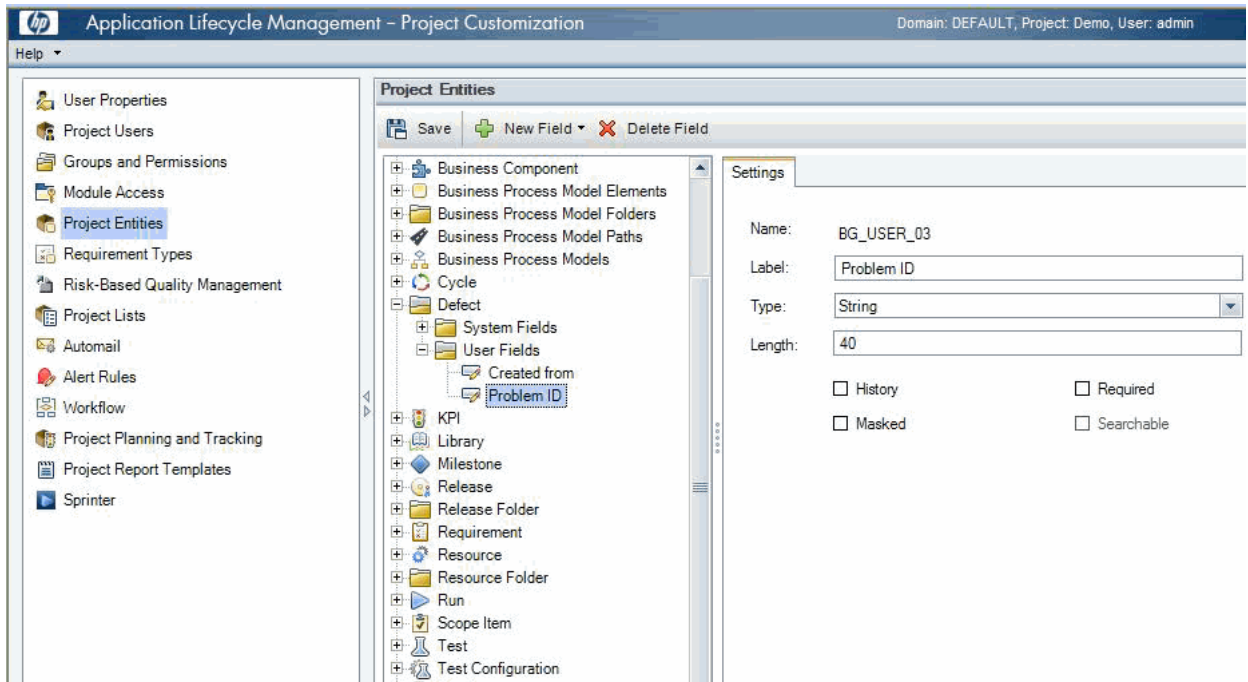
To add the required fields for Defect module customization:

1. Log on to ALM as a project administrator.
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
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](#).

- Click **Workflow > Script Editor**.
- 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
```
- 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 124](#)

### 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 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:

Mapped Fields			
Type	QC Field	Direction	SM ChangeManagement Field
	Severity	<---->	Urgency
	Problem ID	<---->	ProblemID
	Defect ID	---->	QCEntityID
	Summary	<---->	Description
	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, status drive the business process of Help Desk in Process Designer Content Pack 9.30.3 and SM 9.4x Codeless. 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:

The image shows two side-by-side configuration windows. The left window is titled 'QC' and the right window is titled 'SM ProblemManagement'. Both windows have three sections: 'Creation', 'Update', and 'Deletion (Full Synchronization Only)'. Each section contains a question 'When a record is created/updated/deleted in this endpoint...' followed by three radio button options. In the 'QC' window, the 'Do nothing' option is selected for Creation and Update, and 'Do nothing' is selected for Deletion. In the 'SM ProblemManagement' window, the 'Create a corresponding record in the other endpoint' option is selected for Creation, 'Update its corresponding record in the other endpoint' is selected for Update, and 'Do nothing' is selected for Deletion. Red rectangles highlight the selected options in both windows.

Section	QC Option	SM ProblemManagement Option
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 (Full Synchronization Only)	Do nothing	Do nothing

## 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**.

**Problem Control - Problem Identification and Classification**

Record Number:  Status:   
Expected Resolution Date:

Brief Description:




[Classification](#) [Activities](#) [Attachments](#) [Related Records](#) [History](#) [Workflow](#) [QC Integration](#)

Synchronize with QC:

Defect ID:



Server/Domain/Project:

4. Synchronize.

 Cancel Current Task  View Report  Refresh Progress ☒ Auto Refresh

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

5. View the Problem in SM.

**Problem Control - Problem Identification and Classification**

Record Number:  Status:   
Expected Resolution Date:

Brief Description:

[Classification](#) [Activities](#) [Attachments](#) [Related Records](#) [History](#) [Workflow](#) [QC Integration](#)

Synchronize with QC:

Defect ID:

Server/Domain/Project:

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

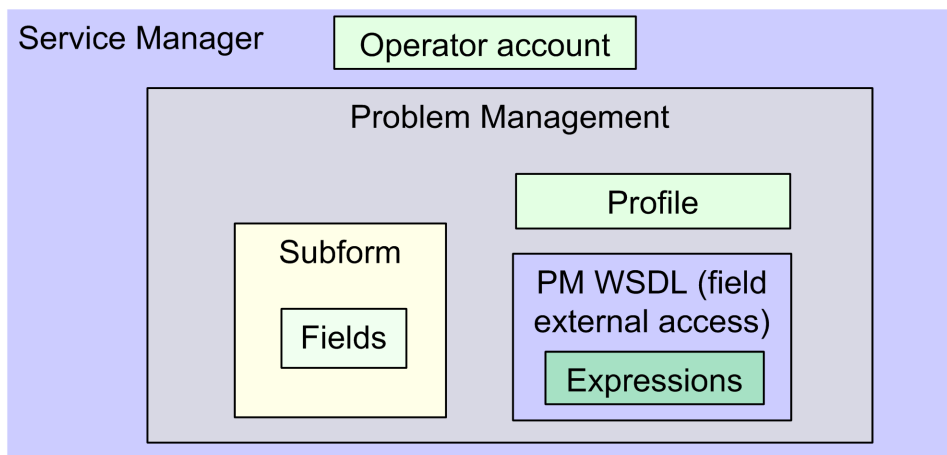
This chapter describes how to synchronize QC/ALM Defects with SM Problems ("Defect -> Problem ").

This chapter includes:

- ["Customizing Service Manager/ServiceCenter for Problem Management" below](#)
- ["Customizing the QC/ALM Defects Module" on page 139](#)
- ["Configuring Links in QC/ALM Synchronizer" on page 147](#)

### Customizing Service Manager/ServiceCenter for Problem Management

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



To customize Service Manager/ServiceCenter 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](#) or ["Specify the External Access Definition on ServiceCenter" on page 131](#)

3. ["Create a Subform" on page 136](#)
4. ["Add the Subform to a Form" on page 137](#)

## Add Fields

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

Field	Type	
	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:

1. Create a custom External Access Definition `QCIntProblemService` by clicking **Tailoring > WSDL configuration** on Service Manager 7.0x; or clicking **Tailoring > Web Services > WSDL 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:

Name: 
Object Name:

☒ Allowed Actions
☒ Expressions
☒ Fields

Allowed Actions	Action Names	Action Type
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	Type
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 Definition

Service Name: QCIntProblemService

Name: rootcause

Allowed Actions

Expressions

Fields

Field	Caption	Type
qcintegration.id	QCEntityID	IntType
id	ProblemID	StringType
sysmodtime	Modified	DateTimeType
qcintegration.project	QCProject	StringType
incident.category	Category	StringType
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 caption value must be unique and alphanumeric (no spaces) with the first letter capitalized (AValidCaption123, AnotherValidCaption, and so on). The above values are required (Do NOT change 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.
- 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)

No	Expression
2	\$rc.update=update in \$L.file;if (denu(\$rc.update)={}) then (\$rc.update={"QC update sent"})
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

Expressions

Fields

Expressions

cleanup(\$pm.activity);cleanup(\$rc.update);if same(update in \$L.file, update in \$L.file.save) then (\$L.need.to.update=true)

\$rc.update=update in \$L.file;if (denu(\$rc.update)={}) then (\$rc.update={"QC update sent"})

if (\$L.need.to.update=true) then (\$rc.update={"QC update sent"})

update in \$L.file=update in \$L.file.save

current.phase in \$L.file="Problem Investigation and Diagnosis"

category in \$L.file="ITIL"

## Specify the External Access Definition on ServiceCenter

To specify the External Access Definition on ServiceCenter:

1. Click **System Defintion > Tables > rootcause > Fields and keys definitions for the rootcause table** and change the settings of these fields.

No.	Field	Include in API	Field name in API	Field data type in API	Remarks
1	id	Y	ProblemID	StringType	
2	sysmodtime	Y	Modified	DateTimeType	
3	qcintegration.id	Y	QCEntityID	IntType	
4	qcintegration.project	Y	QCProject	StringType	
5	qcintegration.type	Y	QCIntegrationType	StringType	
6	qcintegration.created.from	Y	CreatedFrom	StringType	
7	current.phase	Y	CurrentPhase	StringType	
8	category	Y	WorkflowType	StringType	

**Note:** The caption value must be unique and alphanumeric (no spaces) with the first letter capitalized (AValidCaption123, AnotherValidCaption, and so on).

2. Create a new Process record named `rca.qcupdate`.
  - a. Click **Utilities > Tools > Document Engine > Process**. Type Process Name `rca.save` and then click **Search**.
  - b. Change the name to `rca.qcupdate` and click **Add**.

- c. Append the following lines to the Initial Expressions tab, and click **Save**.

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

**Process Definition**

Process Name:

☐ Save Cursor Position? ☐ Run Standard Process when complete?

☐ Run in Window? Window Title:

Initial Expressions Initial Javascript RAD Final Expressions Final Javascript Next Process

```

if same(nullsub(full.name in $G.rc.environment, full.name in $G.rc.global.environment), true) then ($L.operator=nullsub($lo.ufname, nullsu...
$L.stamp=str(tod())+" ("+$L.operator+");"
if ($rc.update={} or $rc.update={""}) then ($rc.update=NULL);if ($kne.update={} or $kne.update={""}) then ($kne.update=NULL);if ($p...
if (update in $L.file=NULL) then (update in $L.file={""})
if ($G.bg and not null($G.bg.activity.type)) then ($rc.update=nullsub($rc.update, $G.bg.activity.text);$rc.update=nullsub($rc.update, "E...
if (filename($L.file)~="rootcausestask") then if (status in $L.file="Past Due" and expected.resolution.time in $L.file>tod()) then (status in $L...

$L.save.status=status in $L.file
if (status in $L.file="Open" or status in $L.file="Reopened" and $reopen.flag=false) then (status in $L.file="Updated")
$reopen.flag=false

cleanup($pm.activity);cleanup($rc.update);if same(update in $L.file, update in $L.file.save) then ($L.need.to.update=true)
$rc.update=update in $L.file;if (dennull($rc.update)={}) then ($rc.update={"QC update sent"})
if ($L.need.to.update=true) then ($rc.update={"QC update sent"})
update in $L.file=update in $L.file.save

```

3. Update the `rca.view` state record.

- a. Add the following row to the `Non-base methods` table, and then click **Save**.

Display Action	Process Name	Condition	Save First
qcupdate	rca.qcupdate	\$L.mode~="close" and \$L.mode~#"add"	false

4. Update the Problem Management integration web service.



- a. Click **Toolkit > WSDL Configuration**. In the Name field type `rootcause`, and then click **Search**.




- b. Update the External Access Definition as follows based on table rootcause.

No.	Field	Value
1	Service Name	QCIntProblemService
2	Object Name	QCIntProblem
3	Allowed Actions/ Action Names	add/Create qcupdate/Update

**External Access Definition**

Service Name:

Name:    Object Name:

 Allowed Actions  Expressions  Data Policy

Allowed Actions	Action Names
add	Create
qcupdate	Update

**Note:** Delete all Allowed Actions without an Action Name.

## Create a Subform

To create a subform:

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	<b>Caption:</b> Defect ID:
2	Text	<ul style="list-style-type: none"><li>• <b>Input:</b> qcintegration.id</li><li>• <b>Read-Only:</b> Yes</li></ul>
3	Label	<b>Caption:</b> Server/Domain/Project:
4	Text	<ul style="list-style-type: none"><li>• <b>Input:</b> qcintegration.project</li><li>• <b>Read-Only:</b> Yes</li></ul>
5	Label	<b>Caption:</b> Created from:
6	Text	<ul style="list-style-type: none"><li>• <b>Input:</b> qcintegration.created.from</li><li>• <b>Read-Only:</b> Yes</li></ul>

The screenshot shows the Forms Designer application window titled "Forms Designer: pm.qcint.subform". The window has a menu bar with "Main Menu: falcon" and a toolbar with "OK", "Cancel", "Delete", and "Design" buttons. The main design area contains three labels and three corresponding text input fields:

- Label: "Defect ID:" followed by a text input field.
- Label: "Server/Domain/Project:" followed by a text input field.
- Label: "Created from:" followed by a text input field.

The HP logo is visible in the bottom right corner of the design area.

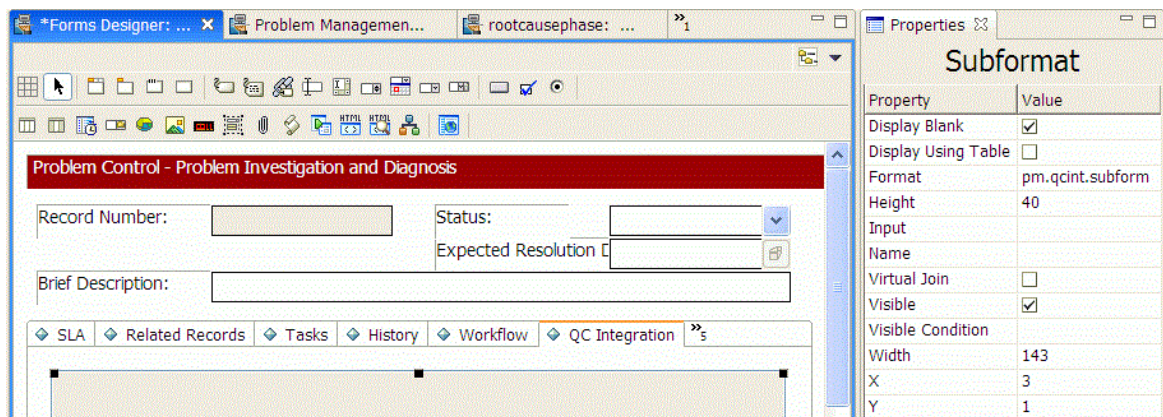


## Add the Subform to a Form

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.

1. 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:



4. Save the changes.

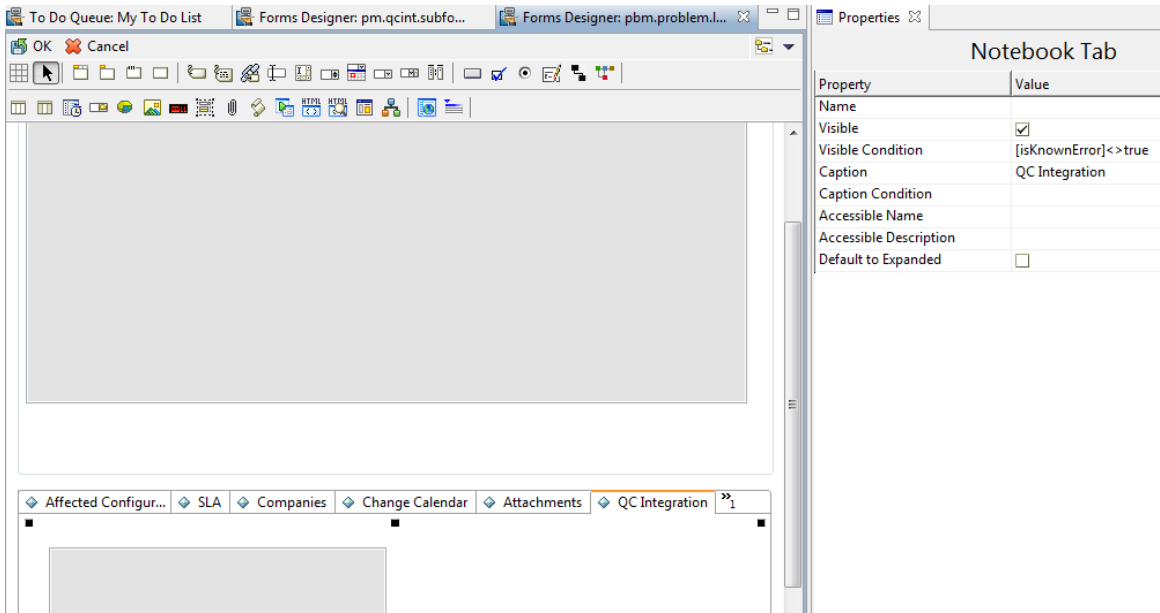
**Note:** If the error message "Format 'pm.qcint.subform' not found (display, show.rio)" appears, restart the SM server to enable the subform.

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:

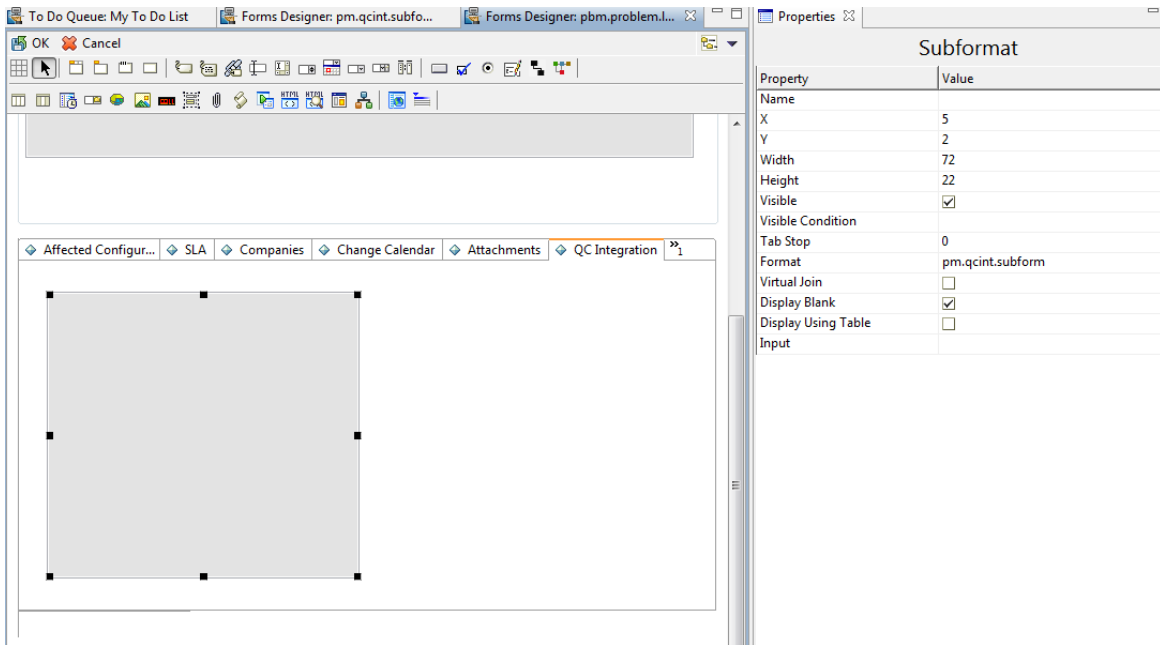
**Note:** The following steps also apply to SM 9.4x Codeless.

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 and then set the value of Visible Condition to [isKnownError]<>true.



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



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 144](#)

### 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 141](#)
4. ["Create a View" on page 143](#)
5. ["Verify" on page 144](#)

### 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	Synchronize with SM Problem	Lookup List/YesNo	Select the "Verify Value" check box
BG_USER_XY	Problem ID	String	

**Quality Center - Project Customization**

**Project Entities**

Project Entities: Defect

- System Fields
- User Fields
  - Change id
  - Forward as problem**
  - Problem ID

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**

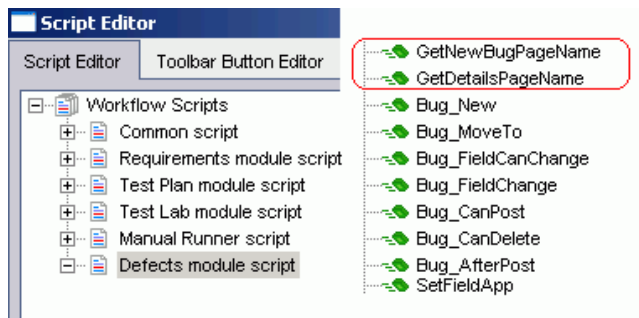
**Workflow**

[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.

[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**.



3. 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**.



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

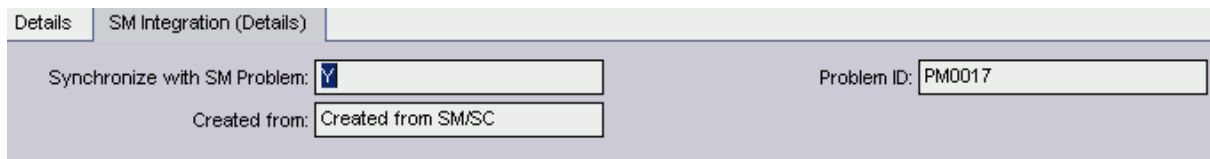


The screenshot shows a dialog box titled "Add Favorite". It has a close button (X) in the top right corner. Below the title bar, there is a label "Add Favorite" followed by a horizontal line. Below this, there is a "Name:" label followed by a text input field containing "SMIntegrationView". Below the name field, there is a "Location:" label followed by two radio buttons. The first radio button is selected and labeled "Private". The second radio button is labeled "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:



The screenshot shows a software interface with a tab labeled "Details" and "SM Integration (Details)". Below the tabs, there are three fields: "Synchronize with SM Problem:" with a dropdown menu set to "Y", "Problem ID:" with a text input field containing "PM0017", and "Created from:" with a dropdown menu set to "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 146](#)
3. ["Verify" above](#)

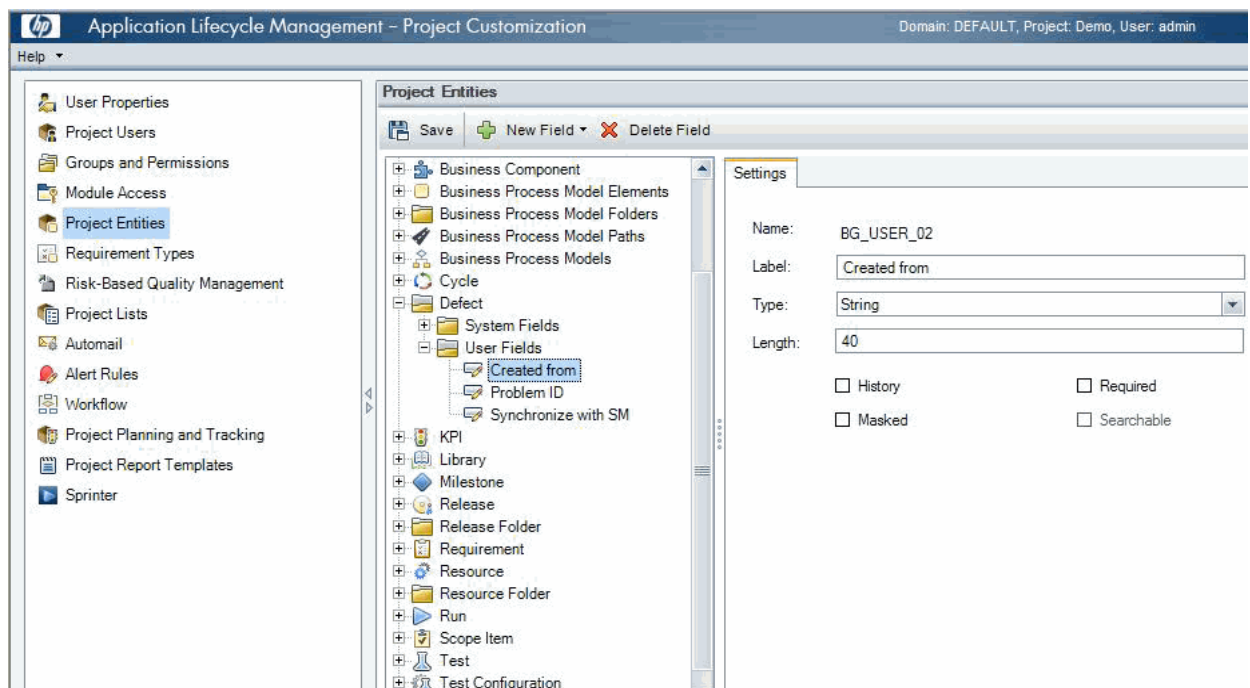
## Add Fields

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_XX	Synchronize with SM Problem	Lookup List/YesNo	Select the "Verify Value" check box
BG_USER_XY	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**.

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 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



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:

The screenshot shows the 'Details' tab of a defect form. The form is divided into two columns. The left column contains fields for 'Actual Fix Time', 'Closing Date', '\* Detected By' (set to 'SMQCIntUser'), 'Detected in Release', 'Detected in Version', 'Planned Closing Ver...', 'Project', 'Assigned To', 'Status' (set to 'New'), 'Target Cycle', 'Change ID', and 'Problem ID' (circled). The right column contains fields for 'Synchronize with SM' (set to 'Y' and circled), 'Closed in Version', 'Detected in Cycle', '\* Detected on Date' (set to '2010-11-12'), 'Estimated Fix Time', 'Priority', 'Reproducible' (set to 'Y'), '\* Severity' (set to '2-Medium'), 'Subject', 'Target Release', 'Created from' (circled), and '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 154](#)
- ["Test the Link" on page 154](#)

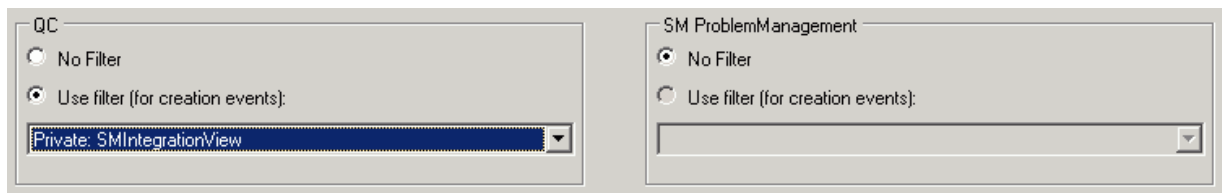
## Specify Endpoints / Type of Link

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 143](#).



















## Define Field Mappings

Basic field mappings are summarized below:

















QC	Direction	SM	Constant Value	Remarks
Problem ID	<-	ProblemID		
Defect ID	->	QCEntityID		<b>Synchronize back on create: Yes</b>
		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.

QC	Direction	SM	Constant Value	Remarks
	->	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.
	->	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 Service Manager 7.10.

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

Mapped Fields			
Type	QC Field	Direction	SM ProblemManagement Field
	Summary	<---->	Description
	Defect ID	---->	QCEntityID
	Severity	<---->	Severity
	Problem ID	<---->	ProblemID
	Value: Created from Quality C...	---->	CreatedFrom
	Value: 1	---->	QCIntegrationType
	Value: AUTO	---->	AssignmentGroup
	Value: BOB.HELPDESK	---->	ProblemOwner
	Value: client system	---->	Category
	Value: software	---->	SubCategory
	Value: email client	---->	ProductType
	Value: outlook	---->	ProblemType
	Value: 4 - User	---->	Impact
	Value: Problem Identification ...	---->	CurrentPhase
	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:

Mapped Fields			
Type	HP-ALM Field	Direction	SM-ProblemManagement Field
	ProblemID	<----	ProblemID
	Summary	<---->	Description
	Severity	<---->	Severity
	Defect ID	---->	QCEntityID
	CreatedFrom	<----	<i>Value: SM</i>
	Detected on Date	<----	<i>Value: 11/2/2010</i>
	<i>Value: QC</i>	---->	CreatedFrom
	<i>Value: 1</i>	---->	QCIntegrationType
	<i>Value: 4 - User</i>	---->	Impact
	<i>Value: DummyQCServer/DEF...</i>	---->	QCProject
	<i>Value: Application</i>	---->	AssignmentGroup
	<i>Value: Software</i>	---->	AffectedItem
	<i>Value: Problem Detection, Lo...</i>	---->	CurrentPhase
	<i>Value: problem</i>	---->	Category
	<i>Value: hardware</i>	---->	SubCategory
	<i>Value: hardware failure</i>	---->	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 Defect -> SM Problem 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	xx
255	BG_SUMMARY	String	Summary	String	<->	String	Description	StringType	description	Char	Text	xx
70	BG_SEVERITY	Lookup List	Severity	Single value list	<->	Single value list	Severity	--	severity	Char	Text	40

If your Service Manager has Process Designer (PD) Content Pack is installed, or if your are working with SM 9.4x Codeless, status drive the business process of Help Desk in Process Designer Content Pack 9.30.3 and in SM 9.4x Codeless. 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, see the following summary for sample constant field mappings:

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

### Constant -> SM Problem Mappings

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, 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, see the table above plus the following row for sample constant field mappings:

QCS SM constant value	Dir	QCS Name/SM WSDL Caption	SM WSDL Type	SM 9.30 DB type	SM DB Name / SM WSDL Field	SM Len
true	->	IsKnownError	BooleanType	char	isKnownError	1

## 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:

The screenshot shows a configuration window with tabs: General, Connectivity, Scheduling, Filters, Events (selected), Field Mapping, and Advanced. The main area is divided into two panels: QC and SM ProblemManagement.

**QC Panel:**

- Creation:** When a record is created in this endpoint...
  - ☒ Create a corresponding record in the other endpoint
  - ☐ Do nothing
- Update:** When a record is updated in this endpoint...
  - ☒ Update its corresponding record in the other endpoint
  - ☐ Do nothing
- Deletion (Full Synchronization Only):** When a record is deleted from this endpoint...
  - ☒ Do nothing
  - ☐ Delete its corresponding record in the other endpoint
  - ☐ Recreate based on its corresponding record in the other endpoint

**SM ProblemManagement Panel:**

- Creation:** When a record is created in this endpoint...
  - ☐ Create a corresponding record in the other endpoint
  - ☒ Do nothing
- Update:** When a record is updated in this endpoint...
  - ☒ Update its corresponding record in the other endpoint
  - ☐ Do nothing
- Deletion (Full Synchronization Only):** When a record is deleted from this endpoint...
  - ☒ Do nothing
  - ☐ Delete its corresponding record in the other endpoint
  - ☐ Recreate based on its corresponding record in the other endpoint

## Test the Link

To test the link:

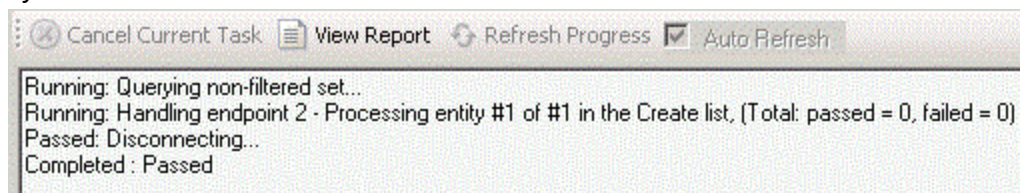
**Note:**

1. 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**.

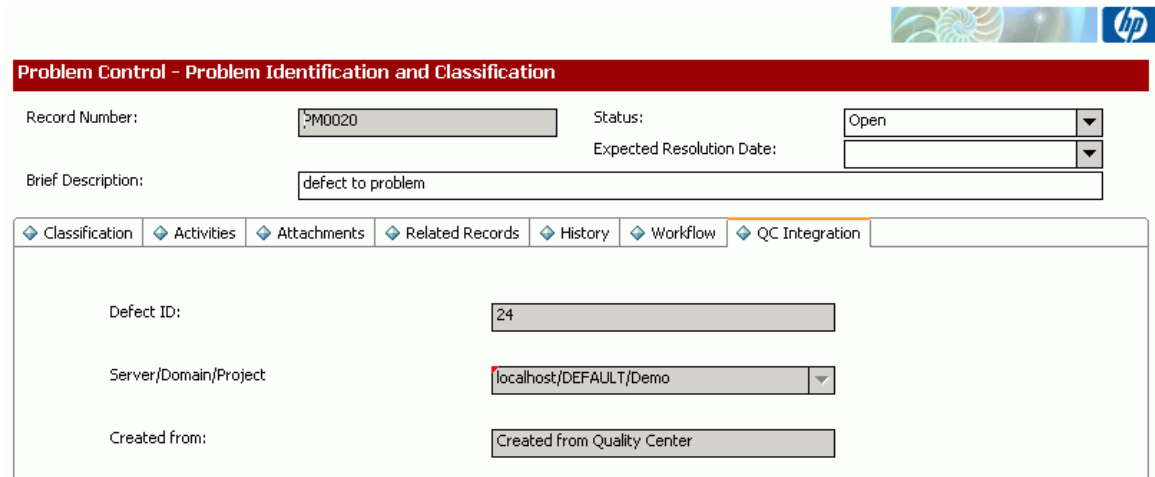
The screenshot shows the 'New Defect' form. The 'Summary' field is filled with 'defect to problem'. The 'Details' tab is active, and the 'SM Integration (New)' section is visible. The 'Synchronize with SM Problem' dropdown is set to 'Y', and the 'Problem ID' field is empty. The form has a 'Submit' button and a 'Close' button at the bottom.

**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.



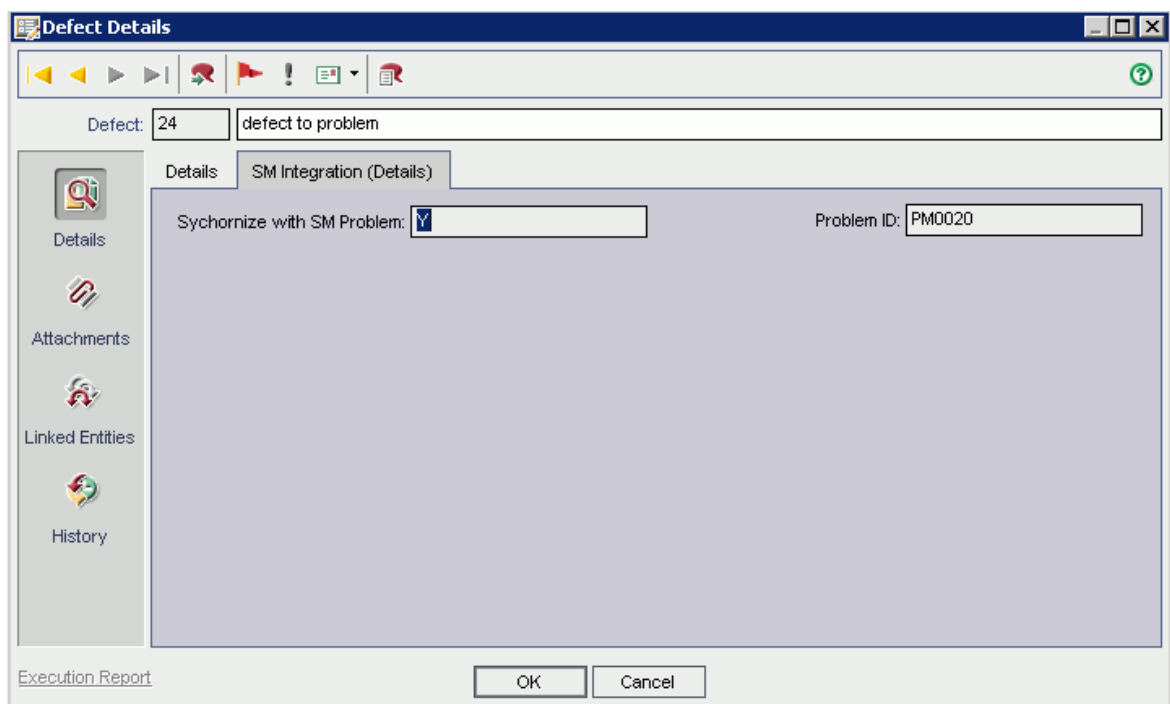
Problem Control - Problem Identification and Classification

Record Number: PM0020 Status: Open  
Expected Resolution Date:  
Brief Description: defect to problem

Classification Activities Attachments Related Records History Workflow QC Integration

Defect ID: 24  
Server/Domain/Project: localhost/DEFAULT/Demo  
Created from: Created from Quality Center

6. View the Defect in QC.



Defect Details

Defect: 24 defect to problem

Details SM Integration (Details)

Synchronize with SM Problem: ☒ Problem ID: PM0020

Details Attachments 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

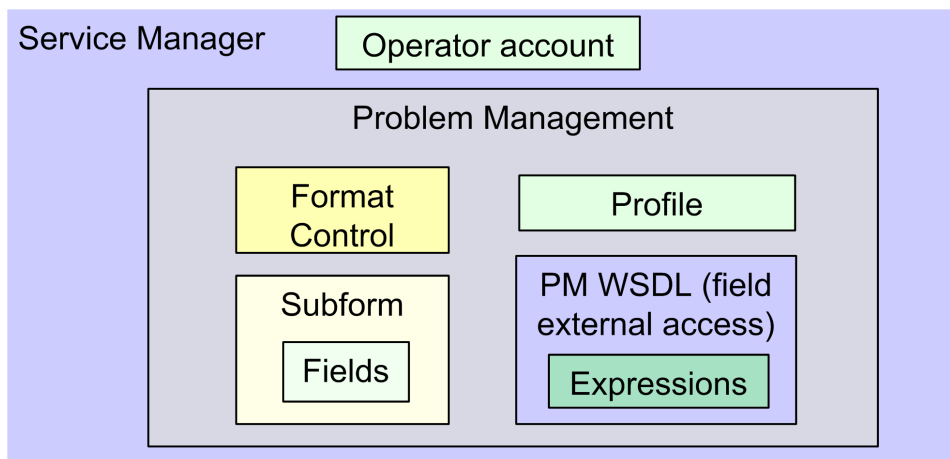
This chapter describes how to synchronize SM Problems with QC/ALM Defects ("Problem <--> Defect").

This chapter includes:

- ["Customizing Service Manager/ServiceCenter for Problem Management "](#) below
- ["Customizing the QC/ALM Defects Module" on page 177](#)
- ["Configuring Links in QC/ALM Synchronizer" on page 186](#)

### Customizing Service Manager/ServiceCenter for Problem Management

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



To customize Service Manager/ServiceCenter 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](#) or ["Specify the External Access Definition on ServiceCenter" on page 163](#)
3. ["Create a Subform" on page 165](#)

4. ["Add the Subform to a Form" on page 168](#)
5. ["Add Format Control Calculations/Validations" on page 169](#)

## Add Fields

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

Field	Type	
	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.

1. Create a custom External Access Definition `QCIntProblemService` by clicking **Tailoring > WSDL configuration** on Service Manager 7.0x; or clicking **Tailoring > Web Services > WSDL 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:

Name:  ▼
 Object Name:

◆ Allowed Actions
◆ Expressions
◆ Fields

Allowed Actions	Action Names	Action Type
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	Type
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 Definition

Service Name: QCIntProblemService

Name: rootcause

Allowed Actions

Expressions

Fields

Field	Caption	Type
qcintegration.id	QCEntityID	IntType
id	ProblemID	StringType
sysmodtime	Modified	DateTimeType
qcintegration.project	QCProject	StringType
incident.category	Category	StringType
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 caption value must be unique and alphanumeric (no spaces) with the first letter capitalized (AValidCaption123, AnotherValidCaption, and so on). The above values are required (Do NOT change 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.
- 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)



No	Expression
2	<code>\$rc.update=update in \$L.file;if (denu(\$rc.update)={}) then (\$rc.update={"QC update sent"})</code>
3	<code>if (\$L.need.to.update=true) then (\$rc.update={"QC update sent"})</code>
4	<code>update in \$L.file=update in \$L.file.save</code>

External Access Definition

Service Name:

QCIntProblemService

Name:

rootcause

Object Name:

QCIntProblem

Allowed Actions

Expressions

Fields

Expressions

```

cleanup($pm.activity);cleanup($rc.update);if same(update in $L.file, update in $L.file.save) then ($L.need.to.update=true)
$rc.update=update in $L.file;if (denu($rc.update)={}) then ($rc.update={"QC update sent"})
if ($L.need.to.update=true) then ($rc.update={"QC update sent"})
update in $L.file=update in $L.file.save
current.phase in $L.file="Problem 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.

1. Refer to [step 1](#).
2. Enable the required fields in the web service.

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

### External Access Definition

Service Name: \* QCIntProblemService ☐ Released:  
 Name: \* rootcause ☐ Deprecated:  
 Object Name: QCIntProblem

Allowed Actions Expressions **Fields**

Field	Caption	Type
id	ProblemID	StringType
sysmodtime	Modified	DateTimeType
qcintegration.id	QCEntityID	IntType
qcintegration.project	QCProject	StringType
qcintegration.type	QCIntegrationType	StringType
qcintegration.created.from	CreatedFrom	StringType

**Note:** The caption value must be unique and alphanumeric (no spaces) with the first letter capitalized (AValidCaption123, AnotherValidCaption, and so on). The above values are required (Do NOT change 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	\$rc.update=update in \$L.file;if (denull(\$rc.update)={}) then (\$rc.update={"QC update sent"})
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 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

```
cleanup($pm.activity);cleanup($src.update);if same(update in $L.file, update in $L.file.save) then ($L.need.to.update=true)
$src.update=update in $L.file;if (dennull($src.update)={}) then ($src.update={"QC update sent"})
if ($L.need.to.update=true) then ($src.update={"QC update sent"})
update in $L.file=update in $L.file.save
isKnownError in $L.file=nullsub(isKnownError in $L.file,true)
```

## Specify the External Access Definition on ServiceCenter

To specify the External Access Definition on ServiceCenter:

1. Click **System Definition > Tables > rootcause > Fields and keys definitions for the rootcause table** and change the settings of these fields.

No.	Field	Include in API	Field name in API	Field data type in API	Remarks
1	id	Y	ProblemID	StringType	
2	sysmodtime	Y	Modified	DateTimeType	
3	qcintegration.id	Y	QCEntityID	IntType	
4	qcintegration.project	Y	QCProject	StringType	
5	qcintegration.type	Y	QCIntegrationType	StringType	
6	qcintegration.created.from	Y	CreatedFrom	StringType	
7	current.phase	Y	CurrentPhase	StringType	
8	category	Y	WorkflowType	StringType	

**Note:** The caption value must be unique and alphanumeric (no spaces) with the first letter

capitalized (AValidCaption123, AnotherValidCaption, and so on).

2. Create a new Process record named `rca.qcupdate`.
  - a. Click **Utilities > Tools > Document Engine > Process**. Type Process Name `rca.save` and then click **Search**.
  - b. Change the name to `rca.qcupdate` and click **Add**.
  - c. Append the following lines to the Initial Expressions tab, and click **Save**.

No.	Expression
1	<code>cleanup(\$pm.activity);cleanup(\$rc.update);if same(update in \$L.file, update in \$L.file.save) then (\$L.need.to.update=true)</code>
2	<code>\$rc.update=update in \$L.file;if (denu(\$rc.update)={}) then (\$rc.update={"QC update sent"})</code>
3	<code>if (\$L.need.to.update=true) then (\$rc.update={"QC update sent"})</code>
4	<code>update in \$L.file=update in \$L.file.save</code>

#### Process Definition

Process Name:

☐ Save Cursor Position? ☐ Run Standard Process when complete?

☐ Run in Window? Window Title:

Initial Expressions Initial Javascript RAD Final Expressions Final Javascript Next Process

```

if same(nullsub(full.name in $G.rc.environment, full.name in $G.rc.global.environment), true) then ($L.operator=nullsub($lo.ufname, nullsu...
$L.stamp=str(tod())+" ("+$L.operator+");"
if ($rc.update={} or $rc.update={""}) then ($rc.update=NULL);if ($kne.update={} or $kne.update={""}) then ($kne.update=NULL);if ($p...
if (update in $L.file=NULL) then (update in $L.file={""})
if ($G.bg and not null($G.bg.activity.type)) then ($rc.update=nullsub($rc.update, $G.bg.activity.text);$rc.update=nullsub($rc.update, "E...
if (filename($L.file)~="rootcausetask") then if (status in $L.file="Past Due" and expected.resolution.time in $L.file>tod()) then (status in $L...

$L.save.status=status in $L.file
if (status in $L.file="Open" or status in $L.file="Reopened" and $reopen.flag=false) then (status in $L.file="Updated")
$reopen.flag=false

cleanup($pm.activity);cleanup($rc.update);if same(update in $L.file, update in $L.file.save) then ($L.need.to.update=true)
$rc.update=update in $L.file;if (denu($rc.update)={}) then ($rc.update={"QC update sent"})
if ($L.need.to.update=true) then ($rc.update={"QC update sent"})
update in $L.file=update in $L.file.save

```

3. Update the `rca.view` state record.

- a. Click **Utilities > Tools > Document Engine > States**. Type the state name `rca.view` and click **Search**.
- b. Add the following row to the Non-base methods table, and then click **Save**.

Display Action	Process Name	Condition	Save First
qcupdate	rca.qcupdate	\$L.mode~="close" and \$L.mode~#"add"	false

4. Update the Problem Management integration web service.

- a. Click **Toolkit > WSDL Configuration**. In the Name field type `rootcause`, and then click **Search**.
- b. Update the External Access Definition as follows based on table `rootcause`.

No.	Field	Value
1	Service Name	QCIntProblemService
2	Object Name	QCIntProblem
3	Allowed Actions/ Action Names	add/Create qcupdate/Update

External Access Definition

Service Name:

QCIntProblemService

Name:

rootcause

Object Name:

QCIntProblem

Allowed Actions

Expressions

Data Policy

Allowed Actions	Action Names
add	Create
qcupdate	Update

**Note:** Delete all Allowed Actions without an Action Name.

## Create a Subform

To create a subform:

1. Create a global list.

- a. Click **Tailoring > Tailoring Tools** on Service Manager 7.0x or later; or click **Utilities > Tools > Global Lists** on ServiceCenter to create 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
6	Value List	{"server1/domain1/project1", "server2/domain2/project2"}	Change to the values for your system  <b>Note:</b> Do not include any spaces between slashes.

- b. Save this global list and click **Rebuild Global List** from the Options menu.

2. 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	<b>Caption:</b> Synchronize with QC

No.	Component	Properties
2	Combo Box	<ul style="list-style-type: none"> <li>■ <b>Input:</b> qcintegration.type</li> <li>■ <b>Value List:</b> 0;1;</li> <li>■ <b>Display List:</b> 0 - Not Synchronize;1 - Synchronize with QC Defect</li> <li>■ <b>Select Only:</b> Yes</li> <li>■ <b>Read-Only Condition:</b> [\$qcint.type.readonly]</li> </ul>
3	Label	<b>Caption:</b> Defect ID:
4	Text	<ul style="list-style-type: none"> <li>■ <b>Input:</b> qcintegration.id</li> <li>■ <b>Read-Only:</b> Yes</li> </ul>
5	Label	<b>Caption:</b> Server/Domain/Project:
6	Combo Box	<ul style="list-style-type: none"> <li>■ <b>Input:</b> qcintegration.project</li> <li>■ <b>Value List:</b> \$G.qcintegration.problem.project</li> <li>■ <b>Read-Only Condition:</b> [\$qcint.project.readonly]</li> <li>■ <b>Mandatory Condition:</b> [qcintegration.type]&gt;0</li> </ul>
7	Label	<b>Caption:</b> Created from:
8	Text	<b>Input:</b> qcintegration.project  <b>Read-Only:</b> Yes

OK Cancel

Synchronize with QC:

Defect ID:

Server/Domain/Project

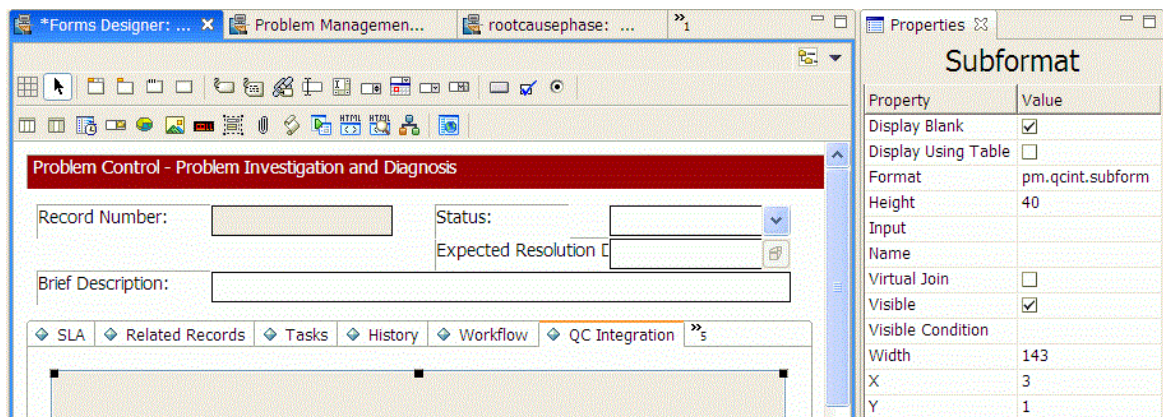
Created from:

## Add the Subform to a Form

If the Service Manager Process Designer (PD) Content Pack is not installed, or you have 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.

1. 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:



4. Save the changes.

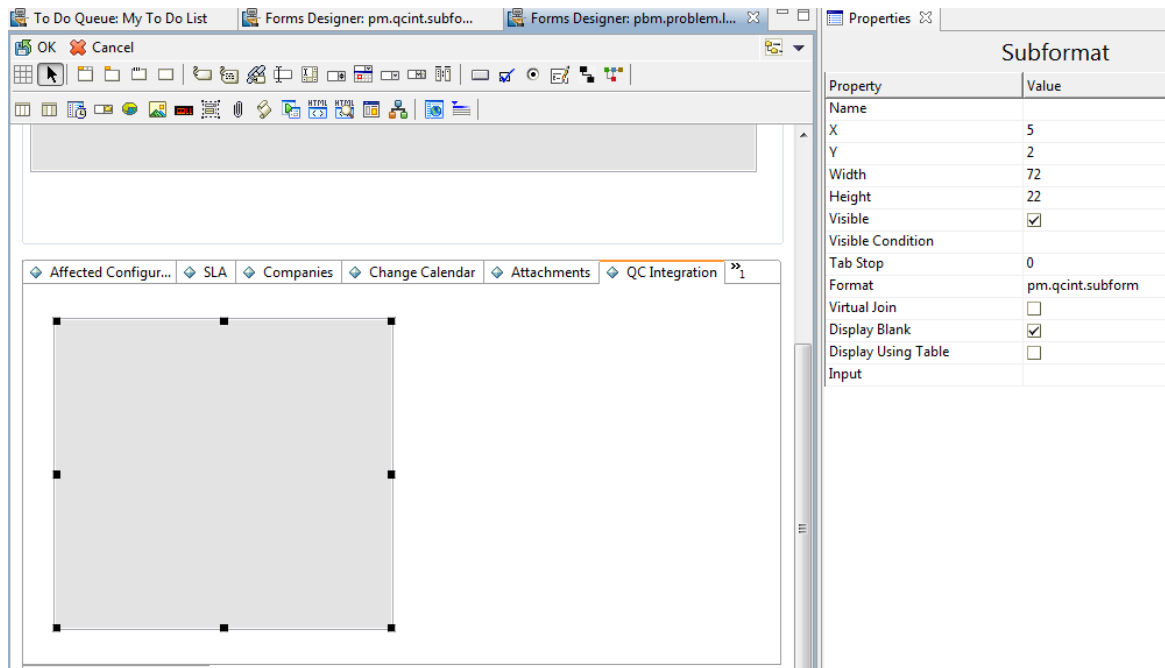
**Note:** If the error message "Format 'pm.qcint.subform' not found (display, show.rio)" appears, restart the SM server to enable the subform.

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.



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:



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 171](#) for SM 9.3x configurations in the PD environment and SM 9.4x Codeless 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	\$qcint.type.readonly=2;if (qcintegration.type in \$file~=0) then (\$qcint.type.readonly=1)
2	display	true
	initial	true
	calculation	\$qcint.project.readonly=2;if (qcintegration.type in \$file~=0 and not null (qcintegration.project in \$file)) then (\$qcint.project.readonly=1)

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
Synchronize with QC:						
						1 - Synchronize with QC Defect ▼
Defect ID:						
						71
Server/Domain/Project						
						localhost/QADEMO/Demo ▼
Created from:						
						Created from Quality Center

## 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 only. Otherwise, refer to ["Add Format Control Calculations/Validations" on page 169](#) 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 173](#)
4. ["Associate the New Workflow with the New Rule Set" on page 175](#)

## Copy an Existing Workflow

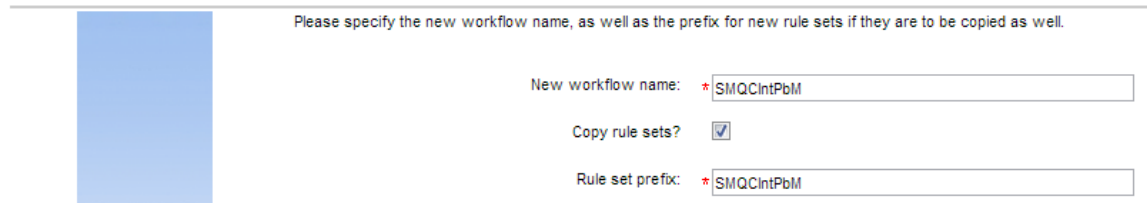
You can use copies of the existing workflows in another business process, or make changes to the HP proprietary workflow copies.

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 prefix for new rule sets if they are to be copied as well.

New workflow name: \* SMQCIntPbM

Copy rule sets? ☒

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:  Apply To:

Active: ☒

Description:

Workflow:

**Subcategories** | **Workflow**

```
graph LR; Logging --> Categorization; Categorization --> Investigation; Investigation --> Resolution; Resolution --> Review; Review --> Closure; Closure --> Abandonment; Abandonment --> Logging; Abandonment --> Categorization; Abandonment --> Investigation; Abandonment --> Resolution; Abandonment --> Review; Abandonment --> 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.

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.functions._null;  var file = vars.\$L_file;  if(file["qcintegration.type"] !=0 &amp;&amp; !_null(file["qcintegration.type"])) {  vars['\$qcint.type.readonly'] = 1  }  if(file["qcintegration.type"] !=0 &amp;&amp; !_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:

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

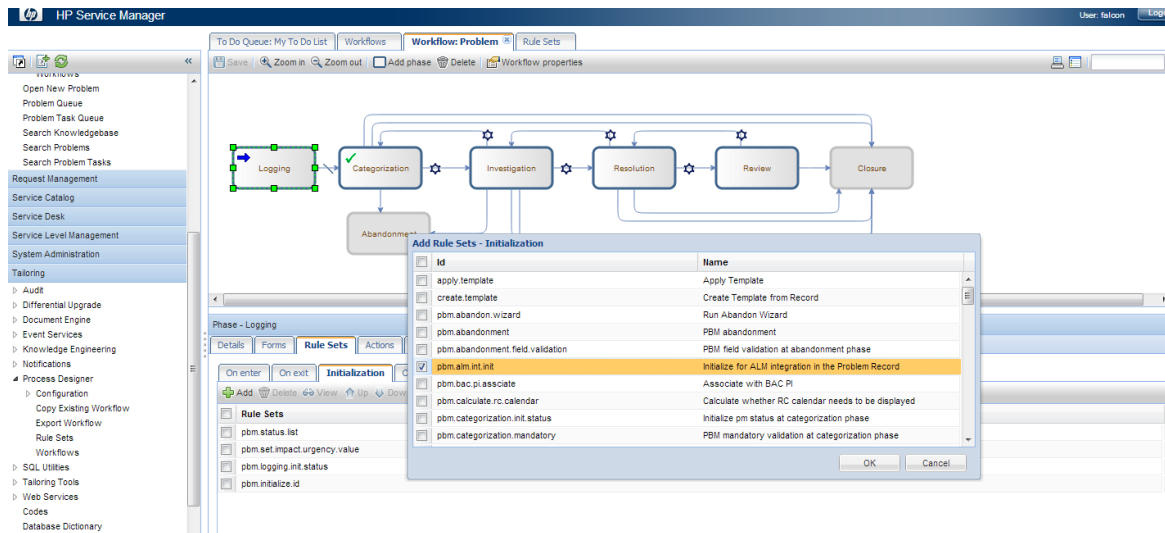
Field	Value
Table name	rootcause
HP Proprietary	

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 61](#).

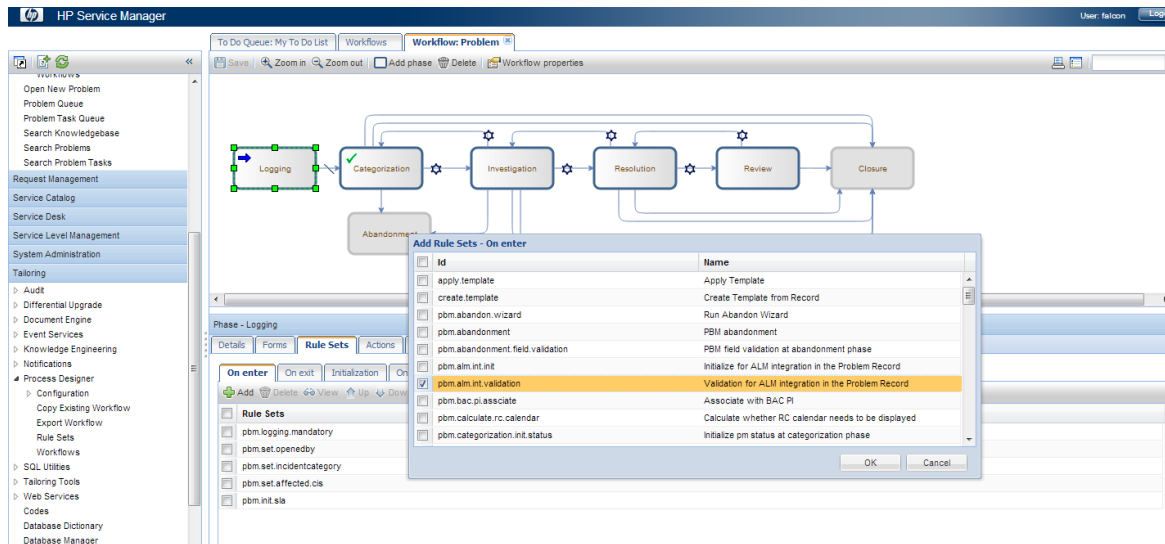
## Associate the New Workflow with the New Rule Set

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.



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.



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 180](#)
4. ["Create a View" on page 182](#)
5. ["Verify" on page 182](#)

### 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_XX	Synchronize with SM Problem	Lookup List/YesNo	Select "Verify Value" check box
BG_USER_XY	Problem ID	String	
BG_USER_XZ	Created from	String	

The screenshot shows the 'Quality Center - Project Customization' window. On the left is a navigation pane with links: User Properties, Project Users, Groups, Module Access, Project Entities (highlighted), Requirement Types, Risk-Based Quality, Project Lists, Automail, Alert Rules, and Workflow. The main area is titled 'Project Entities'. It has two tabs: 'Project Entities' and 'Field Settings'. The 'Project Entities' tab shows a tree structure for the 'Defect' entity, with 'System Fields' and 'User Fields'. Under 'User Fields', there are three items: 'Change id', 'Forward as problem' (highlighted), and 'Problem ID'. The 'Field Settings' tab is active, showing configuration for the selected field. It includes: Field Name: BG\_USER\_01 (circled in red), Field Label: Forward as problem, Field Type: Lookup List (dropdown), Field Length: 40, and checkboxes for History, Required, Masked, and 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**.



3. 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:

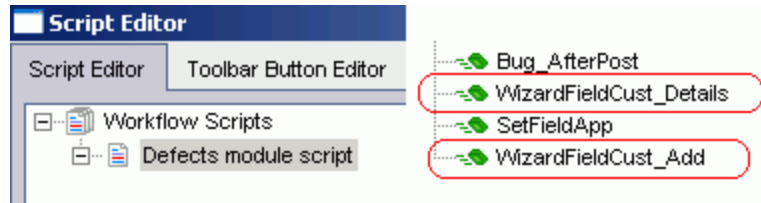
1. In "QC - Project Customization", click **Workflow** > **Script Editor**.
2. Select **Defects module script**.



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

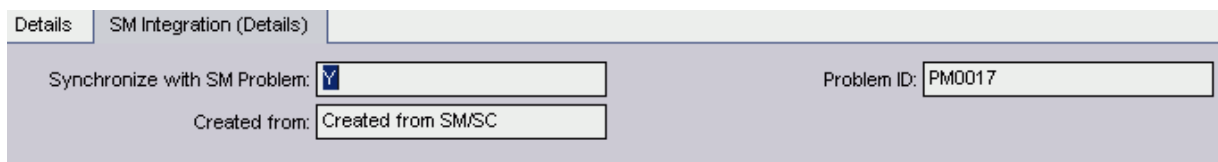
- **Location:** Private



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:



## 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:

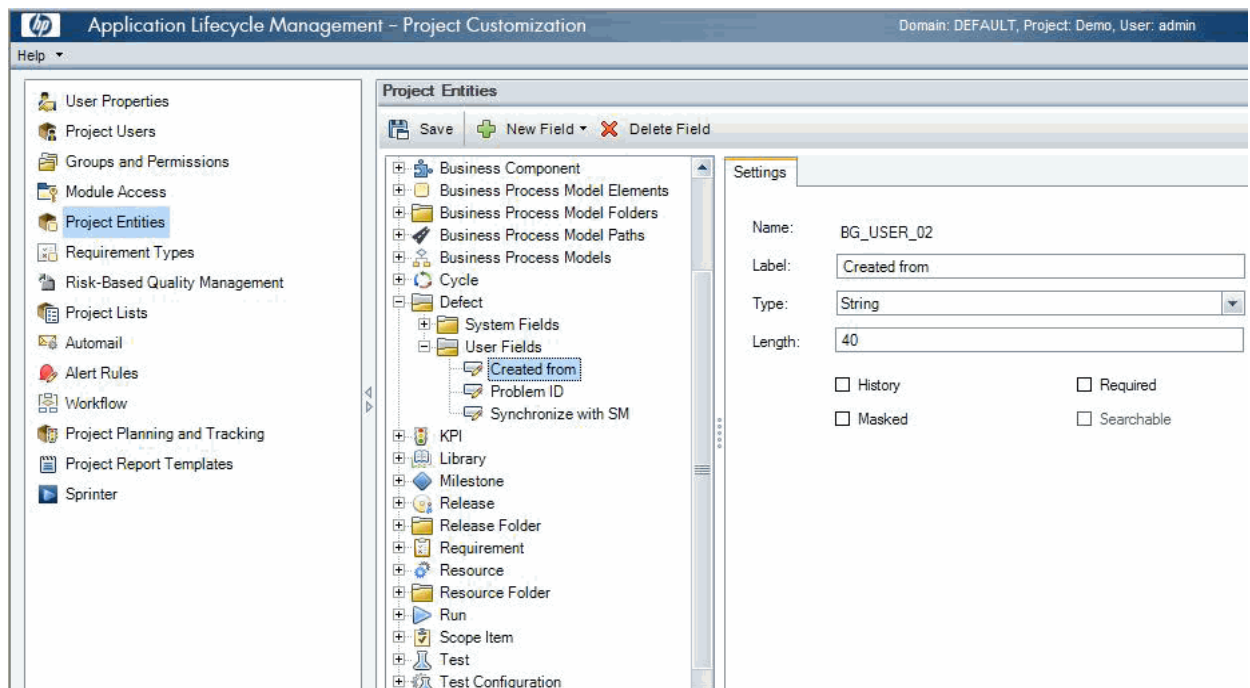
1. ["Add Fields" below](#)
2. ["Create a View" on page 185](#)
3. ["Verify" on page 185](#)

## 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_XX	Synchronize with SM Problem	Lookup List/YesNo	Select the "Verify Value" check box
BG_USER_XY	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**.
6. Set the **Readonly** fields by adding the following lines to the **Bug\_New** and **Bug\_Move** 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



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:	<input type="text"/>
Closing Date:	<input type="text"/>
* Detected By:	SMQCIntUser 
Detected in Release:	<input type="text"/>
Detected in Version:	<input type="text"/>
Planned Closing Ver...	<input type="text"/>
Project:	<input type="text"/>
Assigned To:	<input type="text"/>
Status:	New
Target Cycle:	<input type="text"/>
Change ID:	<input type="text"/>
Problem ID:	<input type="text"/>
Synchronize with SM:	Y
Closed in Version:	<input type="text"/>
Detected in Cycle:	<input type="text"/>
* Detected on Date:	2010-11-12
Estimated Fix Time:	<input type="text"/>
Priority:	<input type="text"/>
Reproducible:	Y
* Severity:	2-Medium
Subject:	<input type="text"/>
Target Release:	<input type="text"/>
Created from:	<input type="text"/>
Modified:	<input type="text"/>

## 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 190](#)
- ["Test the Link" on page 191](#)

### 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**

- 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 143](#).

The screenshot shows two panels for defining filters. The left panel is for 'QC' and the right panel is for 'SM ProblemManagement'. Both panels have a radio button for 'No Filter' and a radio button for 'Use filter (for creation events):'. The 'QC' panel has a dropdown menu open, showing '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	<p>Replace XXX with a valid phase name, such as “Problem Investigation and Diagnosis”.</p> <p>This field mapping is optional for demo data of Service Manager 7.10.</p>
	->	QCProject	(your setup)	<p>This value should be same with the “QC Project” parameter in the Connectivity tab.</p>
	->	WorkFlowType	YYY	<p>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.</p> <p>This field mapping is optional for demo data of Service Manager 7.10.</p>




















Sample field mappings are shown in the following screenshot:

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

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
	->	IsKnownError	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			
Type	QC Field	Direction	SM ProblemManagement Field
	Summary	<---->	Description
	Defect ID	---->	QCEntityID
	Severity	<---->	Severity
	Problem ID	<---->	ProblemID
	Synchronize with SM Problem	<---->	Value: Y
	Value: Created from Quality C...	---->	CreatedFrom
	Value: 1	---->	QCIntegrationType
	Value: AUTO	---->	AssignmentGroup
	Value: BOB.HELPDESK	---->	ProblemOwner
	Value: client system	---->	Category
	Value: software	---->	SubCategory
	Value: email client	---->	ProductType
	Value: outlook	---->	ProblemType
	Value: 4 - User	---->	Impact
	Value: Problem Identification...	---->	CurrentPhase
	Value: localhost/DEFAULT/...	---->	QCProject
	Value: ITIL	---->	WorkflowType
	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:

The screenshot displays a configuration window with two main panels: 'QC' on the left and 'SM ProblemManagement' on the right. Each panel contains three sections: 'Creation', 'Update', and 'Deletion (Full Synchronization Only)'. In each section, there are radio buttons for different actions. In the 'QC' panel, the selected options are 'Create a corresponding record in the other endpoint' for Creation and Update, and 'Do nothing' for Deletion. In the 'SM ProblemManagement' panel, the selected options are 'Create a corresponding record in the other endpoint' for Creation and Update, and 'Do nothing' for Deletion. Red rectangles highlight the selected radio buttons in each section.


## 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".



**Problem Control - Problem Identification and Classification**

Record Number:

PM0021

Status:

Updated

Brief Description:

defect <-> problem new problem

Expected Resolution Date:

Classification

Activities

Attachments

Related Records

History

Workflow

QC Integration

Synchronize with QC:

1 - Synchronize with QC Defect

Defect ID:

Server/Domain/Project

localhost/DEFAULT/Demo


Created from:

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

**New Defect**

Clear

Attach:



\* Summary:

defect <-> problem new defect

Details

SM Integration (New)

Sychornize with SM Problem:

Y

Problem ID:

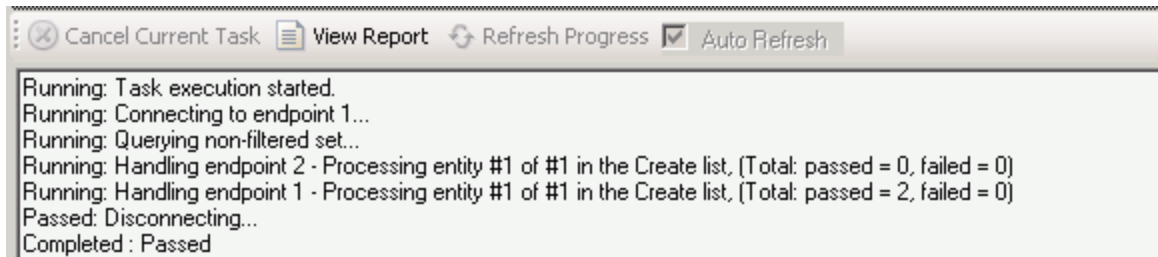
Created from:

Submit

Close



5. Synchronize.



6. View the problem in SM.

**Problem Control - Problem Identification and Classification**

Record Number:  Status:   
Expected Resolution Date:

Brief Description:

Classification Activities Attachments Related Records History Workflow **QC Integration**

Synchronize with QC:

Defect ID:

Server/Domain/Project:

Created from:

---

**Problem Control - Problem Identification and Classification**

Record Number:  Status:   
Expected Resolution Date:

Brief Description:

Classification Activities Attachments Related Records History Workflow **QC Integration**

Synchronize with QC:

Defect ID:

Server/Domain/Project:

Created from:

7. View the defect in QC.

The image displays two screenshots of the 'Defect Details' form in ALM 11, specifically the 'SM Integration (Details)' tab. The top screenshot shows Defect 25 with the title 'defect <-> problem new defect'. The 'Synchronize with SM Problem' checkbox is checked, and the 'Problem ID' is PM0022. The bottom screenshot shows Defect 26 with the title 'defect <-> problem new problem'. The 'Synchronize with SM Problem' checkbox is also checked, and the 'Problem ID' is PM0021. The 'Created from' field in the bottom screenshot is 'Created from SM/SC'. The left sidebar contains links for Details, Attachments, Linked Entities, and 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 205](#)

## 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.
2. 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.
3. 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.
4. 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 202
- "[Upgrade from Non-PD Environment to PD Content Pack 9.30.3/SM 9.4x Codeless](#)" on page 202
- "[Upgrade from 9.3x with PD Content Pack 9.30.2 to 9.30.3/SM 9.4x Codeless](#)" on page 204

## 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 198
- "[SM Problem -> QC/ALM Defect](#)" on page 199

- ["QC/ALM Defect -> SM Problem" on page 199](#)
- ["SM Problem <=> QC/ALM Defect" on page 202](#)

## 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 QC/ALM Customization

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

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

Field Name	Field Label	Field Type	Remarks
BG_USER_XY	Created from	String	

- In the "QC - Project Customization" module, click **Workflow**.
  - Add the following code to the **WizardFieldCust\_Details** and **WizardFieldCust\_Add** event procedures

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

- Set the field to Read-only by adding the following lines to the **Bug\_New** and **Bug\_Move** to subroutines.

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

- 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

- d. 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\_Move to** 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:

Field	Type	
	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 127](#).

- e. Configure WSDL.

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

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

See ["Specify the External Access Definition on Service Manager" on page 127](#) and ["Specify the External Access Definition on ServiceCenter" on page 131](#) 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:

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



- 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

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.

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.

## 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:

1. 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 208](#)
- ["Configuration" on page 208](#)
- ["Runtime" on page 211](#)
- ["XML Validation" on page 215](#)

## 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 Change generation failed.	The Change WSDL URL is not valid.	Provide the valid Change WSDL URL. (For example, <a href="http://localhost:13080/sc62server/PWS/QCIntChangeService.wsdl">http://localhost:13080/sc62server/PWS/QCIntChangeService.wsdl</a> ).
INS_2	The stub for Service Manager Problem generated failed.	The Problem WSDL URL is not valid.	Provide the valid Problem WSDL URL. (For example, <a href="http://localhost:13080/sc62server/PWS/QCIntChangeService.wsdl">http://localhost:13080/sc62server/PWS/QCIntChangeService.wsdl</a> ).

## 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 <code>&lt;QCS_Install_Dir&gt;\adapters\lib</code> .



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 <b>Do nothing</b> 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 <b>Do nothing</b> 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].wsdl

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 <b>Retries on Locked Record</b> in the Advanced tab is out of range (0~3).	Enter an integer (0 ~ 3) for <b>Retries on Locked Record</b> in the Advanced tab.
CFG_13	Retry interval must be an integer between 1 and 10.	The value for parameter <b>Retry Interval(Seconds)</b> in the Advanced tab is out of range (1 to 10).	Enter an integer (1 to 10) for <b>Retry Interval(Seconds)</b> in the Advanced tab.
CFG_14	Socket timeout must be an integer between 0 and 120. (0 means default timeout).	The value for parameter: <b>Socket Timeout (Minutes)</b> in the Advanced tab is out of range (0 to 120).	Enter an integer (0 to 120) for <b>Socket Timeout (Minutes)</b> in the Advanced tab.

## Runtime

MSG_ID	Message	Cause	Solution
RUN_1	Required field <Field Name> can not be empty or SPACE filled.	Synchronized null /space value to a required field from sponsor to receiver.	Ensure that required field values are not null or filled with spaces.
RUN_2	error when reading web service response from SM: 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	<p>There are two possible causes:</p> <ul style="list-style-type: none"> <li>Insufficient rights for the SM Integration user when creating/updating defect/requirement in QC or running synchronization to create/update the corresponding change/problem.</li> <li>Maximum active logins for integration account is exceeded.</li> </ul>	<ol style="list-style-type: none"> <li>Check the rights of integration account.</li> <li>Check and make sure that <b>Unlimited session</b> in the <b>Security</b> tab is selected.</li> </ol>
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>.	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>.	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>.	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>.	Dynamic model cannot find this key.	Ensure that the stub jar has been generated correctly.
RUN_10	Mapping file <file path> cannot be found!	The configuration file isn't found.	Please check the configuration file.
RUN_10	<Module builder class name> cannot be created because of <message>.	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 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>.	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>.	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.	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 <b>DecimalType</b> 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: findRequirementById: Failed getting requirement with id: <id>  Failed to update, record was not found or deleted on target null	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 blank 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 <b>mapping</b> .	Add a root element <b>mapping</b> .
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 <b>module</b> element in the <b>mapping</b> element.	Add a <b>module</b> element in the <b>mapping</b> 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 <b>module</b> elements in the mapping file.	Make sure the <b>mapping</b> element has only one or two <b>module</b> 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 <b>problem</b> or <b>change</b> .	The <b>name</b> attribute of a <b>module</b> element should be <b>change</b> or <b>problem</b> .

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 <b>field</b> element in the <b>module</b> element.	Define <b>field</b> elements in each <b>module</b> 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 <b>field</b> element has a wrong type attribute.	The <b>type</b> attribute of a <b>field</b> 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 <b>field</b> element has a wrong <b>readonly</b> attribute.	The <b>readonly</b> attribute of a <b>field</b> element should be <b>true</b> or <b>false</b> .
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 <b>field</b> element has a wrong <b>required</b> attribute.	The <b>required</b> attribute of a <b>field</b> element should be <b>mandatory</b> , <b>optional</b> or <b>recommended</b> .
XML_9	Failed to validate the configuration file: cvc-complex-type.4: Attribute 'type' must appear on element 'field'.	The <b>field</b> element has no <b>type</b> attribute.	The <b>type</b> attribute must be defined in each <b>field</b> 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 <b>field</b> element has no <b>name</b> attribute.	Define a <b>name</b> attribute in the <b>field</b> 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 <b>field</b> element has more than one child element <b>items</b> .	Define only one <b>items</b> element in each <b>field</b> 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 <b>items</b> element has no child element <b>item</b> .	Add <b>item</b> elements in each <b>items</b> element.
XML_ 13	Failed to validate the configuration file: cvc-complex-type.4: Attribute 'value' must appear on element 'item'.	The <b>item</b> element has no <b>value</b> attribute.	Define a <b>value</b> attribute for each <b>item</b> 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 <b>item</b> element has no <b>text</b> value.	Define a <b>text</b> value for each <b>item</b> 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 <b>value</b> attribute has an empty value.	Define a value for the <b>value</b> attribute in each <b>item</b> 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 <b>name</b> attribute has an empty value.	Define a value for the <b>name</b> attribute of each <b>item</b> 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 <b>field</b> element has a type of <b>Attachment</b> .	Remove the <b>Attachment</b> 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 <b>field</b> element has an incorrect length attribute value.	Correct the value of the <b>field</b> 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 for Process Designer (PD) Content Pack 9.30.2 and 9.30.3)**

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](mailto:ovdoc-itsm@hp.com).

We appreciate your feedback!

