Developer Guide | PUBLIC
2019-09-20

Development Toolkit Guide

SAP Quotation and Underwriting for Insurance 1.1

THE BEST RUN SAP

Content

1 Introduction

1.1 About This Document

This document describes to how to install and configure the SAP Quotation and Underwriting for Insurance (FS-QUO) development toolkit to create an FS-QUO development environment.

1.2 Audience

This document is intended for technical consultants and developers who modify FS-QUO functionality. It's assumed that you have experience developing applications using UI5 and Java.

2 Prerequisites

2.1 Prerequisites for Installing the FS-QUO Toolkit

This topic details the software that is required before you can install the FS-QUO toolkit.

Before you install the toolkit, download the following software:

Oracle Java Development Kit SE8 for Windows x86_64	This is required to run Eclipse. Download the software here .
Eclipse Oxygen IDE for Java EE Developers Windows 64 bit	Download the software here .
Apache Maven 3.2.2	Download the software here .
Apache TomEE+ 1.7.5	Download the software here .
SAP HANA Client 2.0, revision 003	Use the SAPCAR tool to extract the HANA client SAR file, then extract `ngdbc.jar` from `SAP_HANA_CLIENT/client/JDBC.TGZ`. Download the client here You can download the SAPCAR utility here.
SAP Java Connector (JCo) 2.x and 3.x	Download SAP Jco 2.x and 3.x versions by searching for "SAP JCo" in the Software Download Center
SAP Cloud Platform Cloud Connector	Download the Windows 64-bit x86 software available in the Cloud Connector area here.

2.2 Understanding the FS-QUO Development Toolkit Contents

The development environment consists of all the required software and source code to help you extend FS-QUO functionality.

> ⚠ Caution
>
> Before you work with the FS-QUO source code, see the Source Code Terms and Conditions [page 6].

The FS-QUO development toolkit is a ZIP file that contains the pre-configured FS-QUO Maven projects. These projects include the source code.

The following projects are provided in the toolkit:

ASLibrary	This project has been deprecated.

ASModel	This project has been deprecated.
ASTxnModel	This project has been deprecated.
ASWebService	This project has been deprecated.
com.sap.fs.quo.domain.data	The Domain Data Model for FS-QUO.
com.sap.fs.quo.cov.service.impl	Business services for coverage-based products in FS-QUO.
com.sap.fs.quo.cov.service.internal	Core services for coverage-based products in FS-QUO.
com.sap.fs.quo.cov.web	OData services for coverage-based products in FS-QUO.
com.sap.fs.quo.cov.service.api	Public API for the business services for coverage-based products in FS-QUO.
com.sap.fs.quo.service	Business service interface for risk-based products in FS-QUO.
com.sap.fs.quo.service.api	Public API for the business services for risk-based products in FS-QUO.
com.sap.fs.quo.service.impl	Business services for risk-based products in FS-QUO.
com.sap.fs.quo.service.internal	Core services for risk-based products in FS-QUO.
com.sap.fs.quo.web	OData services for risk-based products in FS-QUO.
dependencies	Compiled JAR files and 3rd party libraries.
DPPWeb	The DDPWeb project contains the *Personal Data Search* UI5 application, for use in DPP compliance. The app allows the search for personal data using a personal identifier, exporting of personal data as CSV file and the anonymization of personal data.
FS-QUO-dev	A module containing artifacts, such as a local `<CSI_HOME>`, required to run FS-QUO in the integrated development environment.
FS-QUO-dev-webapp	Artifacts for deployment to an application server. Contains additional artifacts for development and testing.
FS-QUO-install	A module containing the `<CSI_HOME>` content for FS-QUO and creating the `<CSI_HOME>` package for deployment.
FS-QUO-mta	Artifacts for deployment to an SAP HANA XS Advanced environment.
FS-QUO-webapp	Artifacts for deployment to an application server.
IntegrationMessageSpec	This project has been deprecated.
IPWWeb	The web container for the FS-IPW back-end.
MultiLevelRiskRenderer	This project has been deprecated.
QUOLibrary	The `QuoODataWeb` project is a dynamic web project, which exposes FS-QUO back-end functionality and services as OData services. This project is built as a WAR module and assembled into the FS-QUO`.ear` with web context root, 'quo-odata'.
QUOODataLibrary	The `QuoODataWeb` project is a dynamic web project, which exposes FS-QUO back-end functionality and services as OData services.

	This project is built as a WAR module and assembled into the `FS-QUO.ear` with web context root, 'quo-odata'.
UWCMSService	This module contains the Content Management System functionality used by the Underwriter Worklist app. The default implementation uses SAP Mobile Documents for content management and Netweaver VSI for virus scanning of file attachments.
UWLibrary	This project provides business and data access services specific to FS-QUO Underwriting. The business services contained within this module are primarily consumed by the OData processors in UWODataLibrary. This project builds as a JAR UWModel. This module provides the central location for UW bean definitions.
UWModel	This module provides the central location for Underwriting bean definitions.
UWODataLibrary	This project contains the Underwriting specific extensions and implementations of `PQMODataLibrary`. The OData processors defined within this library primarily consume the core FS-QUO and Underwriting business services contained within ASLibrary and UWLibrary, respectively.

2.3 Source Code Terms and Conditions

The FS-QUO source code described in this guide is the intellectual property (IP) of SAP SE. The source code is available for customers to support extension.

Any changes that you make to the core source code voids part of the warranty or the entire warranty.

For more information about IP terms and conditions, see the license agreement.

2.4 Supported Clients

The FS-QUO client is browser-based.

For information about the browsers that are supported, see Client System Requirements.

2.5 Substitution Variable Placeholders

The following table defines the substitution variables used in this guide. Gather the information listed prior to beginning.

Substitution Variable	Description
`<rel>`	The FS-QUO release version that you are installing.
`<sp>`	The support package version you are installing or upgrading.
`<pl>`	The patch level of you are installing or upgrading.
`<CSI_HOME>`	The installation directory of the file system portion of an FS-PRO or FS-QUO installation.
`<eclipse_dir>`	The local directory where the Eclipse application is installed.
`<toolkit_dir>`	The local directory where the FS-QUO toolkit is installed.
`<jdk_dir>`	The installation directory of the JDK.
`<dev_profile_name>`	The name of your FS-QUO development profile.
`<quo_toolkit_url>`	The URL to the FS-QUO Design Time approuter application corresponding to your local FS-QUO development setup - "localhost:8080".
`<FS-PM_dest_name>`	The destination name for connectivity with FS-PM.
`<BP_dest_name>`	The destination name for connectivity with SAP GP-FS.

3 Installing the FS-QUO Toolkit

This topic describes, at a high-level, the steps involved in installing the FS-QUO toolkit.

You need to perform the following steps in order to install the FS-QUO toolkit:

- Connecting to the FS-QUO Runtime Database
- Installing and Configuring Eclipse
- Installing and Configuring TomEE+
- Installing and Configuring the FS-QUO Toolkit
- Copying and Deploying Insurance Product JAR Files from the Product Modeler
- Configuring the FS-QUO Toolkit for FS-QUO
- Installing and Configuring Standalone SAP JCo (Optional)
- Validating the Complete FS-QUO Toolkit Installation

3.1 Connecting to the FS-QUO Runtime Database

Using the steps described in the *SAP Quotation and Underwriting for Insurance Installation Guide*, the solution architect in the FS-QUO development team is responsible for setting up the FS-QUO development infrastructure for the first time in the XSA environment. The FS-QUO database will be shared by the development team, who must have the necessary access to the XSA environment.

The instructions to create the environment are contained in the *SAP Quotation and Underwriting for Insurance Installation Guide*.

> ⓘ Note
>
> Record the JDBC information including the JDBC URL and development database user credentials to configure the FS-QUO toolkit database connection.

3.2 Installing and Configuring Eclipse

This topic details the steps required to install and configure Eclipse for use with the FS-QUO toolkit.

Context

Perform the following steps to install Eclipse:

Procedure

1. Unzip `eclipse-jee-oxygen-3a-win32-x86_64.zip` to a local directory, which will be referred to as `<eclipse_dir>` in this guide.
2. Edit `eclipse.ini` to configure Eclipse to use the JDK as the default VM.
   a. Open `<eclipse_dir>\eclipse.ini` with a text editor.
   b. Add the following lines to the beginning of the file:

      ```
      -vm
      <jdk_dir>\bin\javaw.exe
      ```

   c. Save your changes and close the file in the editor.
3. In *Windows Explorer*, open Eclipse by browsing to `<eclipse_dir>` and running `eclipse.exe`.
4. When prompted to select a workspace, either use the default path or provide a temporary path to open Eclipse.
5. If you are working behind a network proxy, configure Eclipse with the proxy server information as follows:
   a. Select *Window* > *Preferences* in the top menu bar.
   b. Expand the left menu tree to *General* > *Network Connections*.
   c. Select the *Manual proxy configuration* option in the configuration pane and enter the HTTP proxy and SSL proxy details.

      If you don't have this information, contact your system or network administrator.
   d. Choose *OK*.
      Your changes are saved.
6. Install any other Eclipse features, such as for source control, as appropriate for your development environment.

3.3 Installing and Configuring TomEE+

The FS-QUO toolkit uses Apache TomEE+ 1.7.5 as the local application server, which must be installed and configured.

Context

The SAP Java Buildpack, bundled in SAP HANA XS Advanced, provides TomEE as one of the Java runtime environments which FS-QUO uses. For local development, FS-QUO toolkit uses a stand-alone installation of TomEE+. While on SAP HANA XS Advanced the TomEE installation is configured as the FS-QUO application is deployed, the local TomEE installation for the FS-QUO toolkit must be manually configured to adopt to FS-QUO's requirements.

Procedure

1. Unzip `apache-tomee-1.7.5-plus.zip` to a local directory, which will be referred to as `<tomee_dir>` throughout this document.
2. Make a backup copy of the following files in `<tomee_dir>\conf`, both of which will be customized for FS-QUO later:
   - `system.properties`
   - `tomee.xml`
3. Customize `<tomee_dir>\conf\system.properties` as follows:
   a. Open `<tomee_dir>\conf\system.properties` in a text editor.
   b. Add the # character in front of the following line in the file to comment it out:

      ```
      tomee.serialization.class.blacklist = *
      ```

   c. Add the following lines to the bottom of the file:

      ```
      openejb.classloader.forced-load=org.apache.commons
      org.apache.activemq.SERIALIZABLE_PACKAGES=*
      ```

   d. Save your changes and close the file in the text editor.

Results

TomEE+ is now installed and configured for integration with Eclipse for use by the FS-QUO toolkit.

3.4 Installing and Configuring the FS-QUO Toolkit

After the initial Eclipse and TomEE application server setup is complete, you can set up the FS-QUO toolkit as an Eclipse workspace by following the instructions in this section.

3.4.1 Unzipping the FS-QUO Toolkit

The FS-QUO toolkit is provided in the form of an Eclipse workspace directory.

Unzip `FS-QUO-toolkit-<rel>.<sp>.<pl>-backend.zip` to a local directory, referred to as `<toolkit_dir>`.

> **Note**
>
> Don't use a directory containing spaces in its path.

3.4.2 Setting up a Development Profile

Context

To provide a consistent development environment to all developers, the FS-QUO toolkit introduces a concept called development profile. A development profile contains artifacts that are common across the development project, including the following items:

- JDBC driver and data source XML file
- TomEE data source XML file
- FS-QUO configuration files in `<CSI_HOME>`
- Insurance product JAR files in `<CSI_HOME>`

Perform the following steps to set up a development profile:

Procedure

1. Create the new directories `<toolkit_dir>\FS-QUO-toolkit\FS-QUO-dev\src\profiles` and `<toolkit_dir>\FS-QUO-toolkit\FS-QUO-dev\src\profiles\<dev_profile_name>`.
2. Create the following subdirectories under the newly created development profile:
   - `home`

- `server\conf`
- `server\lib`

3. Copy the SAP HANA JDBC driver file (`ngdbc.jar`) that you downloaded as a prerequisite to `<toolkit_dir>\FS-QUO-toolkit\FS-QUO-dev\src\profiles\<dev_profile_name>\server\lib`.
4. Copy the sample TomEE configuration file in `<toolkit_dir>\FS-QUO-toolkit\FS-QUO-dev\src\samples\tomee.xml` to `<toolkit_dir>\FS-QUO-toolkit\FS-QUO-dev\src\profiles\<dev_profile_name>\server\conf`.
5. Open the new `<toolkit_dir>\FS-QUO-toolkit\FS-QUO-dev\src\profiles\<dev_profile_name>\server\conf\tomee.xml` file with a text editor.
6. Replace the placeholder texts surrounded by ### signs with information specific to the FS-QUO Runtime database that you created earlier in the XML file.
7. Save your changes.
8. Close the file in the text editor.

3.4.3 Initializing the FS-QUO Toolkit Workspace in Eclipse

Prerequisites

- You have JDK 1.8 installed
- You have Apache Maven version 3.2.2 installed.

Context

Now the workspace is configured in the file system, perform the following steps to initialize the workspace within Eclipse by importing projects and creating a TomEE server:

Procedure

1. Create a parent directory for the `<toolkit_dir>` directory, which will be your Eclipse workspace location.
2. Open Eclipse and provide the parent directory created in the previous step as the workspace location when prompted.
3. Verify the JRE and compiler settings:
   a. Choose *Window* > *Preferences* from the menu bar.
   b. Select *Java* > *Installed JREs* > *Execution Environments* > *JavaSE-1.8* and check the 1.8 JDK that was previously added.

c. Select *Java* > *Compiler* > *JDK Compliance* from the menu tree and set your compiler level by selecting *1.8* from the dropdown.
d. Select *Maven* > *Installations* in the menu tree.
e. Ensure the *apache-maven-3.2.2* installation is selected.
f. Select *Maven* > *Java EE Integration* in the menu tree.
g. Under *Select active Java EE configurators*, deselect all three configurator options.
h. Choose *OK*.
Your changes are saved.

4. Select *Window* > *Open Perspective* > *Other...* from the menu bar, then select *Java EE* to open the *Java EE* perspective.
5. Import the projects into the workspace as follows:
   a. In the menu bar, select *File* > *Import...*, then select *Maven* > *Existing Maven Projects into Workspace...* and choose *Next*.
   b. Choose *Browse...* for the *Select root directory* field and select `<toolkit_dir>` (which should be selected by default).
   c. Choose *OK*.
   d. Ensure that *Select All* has been chosen by default and choose *Finish*.

> **Note**
>
> Immediately after import, the workspace will contain numerous build path errors. The next few steps will provision Maven dependencies properly and resolve these errors.
>
> There are also errors in the following categories that will persist but can be ignored:
> - WSDL Problem
> - XML Schema Problem
> - XSL Problem

6. In the *Project Explorer* view, select *FS-QUO-toolkit* and choose *Maven* > *Update Project...* from the popup menu. In the new dialog, choose *OK*, then wait for the workspace to be updated and built.
7. In the *Project Explorer* view, select *FS-QUO-toolkit* and choose *Run As...* > *Maven install* from the popup menu.

   Maven will install binary artifacts from the FS-QUO toolkit to your local Maven repository as part of the initialize phase, compiling, running unit tests, and installing artifacts on all projects.

   When the update has completed, ensure that there are no Java problems in the *Markers* view. To remove Java problems 'related to build path errors' after a successful Maven installation, run the Maven Update project again and it should remove those errors.
8. Initialize the `<CSI_HOME>` directory and the development profile as follows:
   a. In the *Project Explorer* view, select FS-QUO-*dev* and choose *Run As...* > *Maven build...* from the popup menu.
   b. Enter **`compile`** in the *Goals* field.
   c. Add the following parameters:

```
tomee.dir: <tomee_dir>
profile: <dev_profile_name>
```

d. Choose *Run* to run the Maven configuration.

9. Perform the following steps to generate an encryption key required by FS-QUO:
   a. Ensure system variable `JAVA_HOME` is set in your system properties to point to a Java directory.
   b. Open the following file with a text editor and change the properties according to the file comments to set a passphrase: `<toolkit_dir>\FS-QUO-toolkit\FS-QUO-dev\target\home\bin\genkey.properties`

      The key will be generated by converting characters to a series of numbers in 0-255 range.

      The passphrase can be of any length and contain any characters (including Unicode). The recommended size is at least 16 characters for a 128-bit key.
   c. Run the `<toolkit_dir>\FS-QUO-toolkit\FS-QUO-dev\target\home\bin\genkey.bat` command and follow the displayed instructions to generate a key.

      The `key.config` file is generated in `<toolkit_dir>\FS_QUO-toolkit\FS-QUO-dev\target\home\system\config`.

10. In the *Project Explorer* view, select FS-QUO-*toolkit* and choose *Maven* > *Update Project...* from the popup menu. In the new dialog, choose *OK*, then wait for the workspace to be updated and built. At this point, all Java build path problems should be resolved.
11. If you don't already have a server defined, create one in the *Servers* view:
    a. Right-click on the *Servers* view area and choose *New* > *Server* from the popup menu.
    b. On the *Define a New Server* wizard page, select *Tomcat v7.0 Server* in the `Apache` folder as the server type and choose *Next*.

       This server type is compatible with TomEE 1.7.

       > **Note**
       >
       > To better identify the server type, you can change the default name *Tomcat v7.0 Server at localhost* to *TomEE+ v1.7.5 Server at localhost* or a similar name for the server.

    c. On the *Tomcat Server* wizard page, enter the `<tomee_dir>` in the *Tomcat installation directory* field and then choose *Next*.
    d. In the *Add and Remove* wizard page, choose *Finish*.
12. Configure the server JVM system properties for FS-QUO as follows:
    a. In the *Servers* view, open the server in the server editor.
    b. Expand *Server Locations* and select *Use Tomcat installation (takes control of Tomcat installation)*.
    c. Expand *Timeout* and increase the value of *Start (in seconds)* to 90, and *Stop (in seconds)* to 30 respectively.
    d. Finally, in the top menu bar, select *File* > *Save* to save the settings.
    e. While still in the *Overview* tab, choose the *Open launch configuration* link in the *General Information* section.
    f. In the *VM arguments* field, ensure the following arguments are present:

       ```
       -DCSI.home=<toolkit_dir>/FS-QUO-toolkit/FS-QUO-dev/target/home
       -Dfile.encoding=UTF-8
       -Dorg.eclipse.emf.ecore.EPackage.Registry.INSTANCE=org.eclipse.emf.ecore.impl.EPackageRegistryImpl
       ```

       Ensure that you use '/' as the file separator for the `CSI.home` value.

13. Choose *OK*.
    Your changes are saved.
14. Close the server editor.
15. Add the EAR module to the TomEE server for deployment.
    a. Right-click *TomEE+ v1.7.5 Server at localhost* in the *Servers* view and choose *Add and Remove...* from the popup menu.
    b. Select *FS-QUO-dev-webapp(FS-QUO-dev-webapp-<rel>.<sp>.<pl>)* from the *Available* list and choose *Add*.
    c. Choose *Finish*.
    Your changes are saved.
    d. Go to the *Modules* tab, select *Edit...* and change the *Path* to `/csiroot`.
    e. Select *OK* to save your change.
    f. Right-click *TomEE+ v1.7.5 Server at localhost* in the *Servers* view and choose *Publish* from the popup menu.

    To verify, open `<tomee_home>\apache-tomee-plus-1.7.5\conf\server.xml` in the editor and go to the end. Ensure that the path has been changed to `/csiroot`. If the change is not present, you may need to restart TomEE multiple times.

3.4.4 Validating the Initial Configuration of the FS-QUO Toolkit

Context

With the initial configuration completed, start the server and verify that the FS-QUO application is running:

Procedure

1. In the *Servers* view, select *TomEE+ v1.7.5 Server at localhost* and choose *Start* from the popup menu.

   Note the port number that will be used to connect to the Administrative Console. The default port for TomEE is 8080.
2. Wait for the server to start.

   When the start process is completed, check the Administrative Console view to ensure that the application starts without errors.
3. Launch Internet Explorer and and log in to the Runtime Administrative Console at the following location: `<quo_toolkit_url>/csiroot/admin/`.
4. Log in using **`admin`** as the user name and **`password`** as the password.

   The application has started successfully if you can access the Runtime Administrative Console.

5. Choose *Logout*.

3.4.5 Configuring the Runtime Administrative Console

Context

Procedure

1. Launch Internet Explorer and log in to the Runtime Administrative Console.
2. Configure the ICM application ID in FS-PM:
   a. Go to ▶ *System* ▶ *Edit Configuration Settings* ▶.
   b. Go to ▶ *Configuration* ▶ *Application* ▶ *AuthoritySuite* ▶ *Env* ▶ *FS-PM Integration Setting* ▶.
   c. Enter value for `icm.application.id` in the *Override* field
   d. Save your changes.
   e. Go to ▶ *System* ▶ *Reload Configuration Settings* ▶
   A list of options appears in the tree panel.
   f. Select the configuration setting type that you want to reload, or choose *Reload All Config* to reload all the settings.
3. If your solution includes an integration with a Content Management System (CMS) you will need to configure the following settings:
   a. Go to ▶ *System* ▶ *Edit Configuration Settings* ▶.
   b. Go to ▶ *Configuration* ▶ *System* ▶ *Env* ▶ *Content Management System Setting* ▶.
   c. Enter values in the *Override* columns for `CMSServerUrRL`, `CMSServerUser` and `CMSServerPassword`.
   d. Save your changes.
   e. Go to ▶ *System* ▶ *Reload Configuration Settings* ▶
   A list of options appears in the tree panel.
   f. Select the configuration setting type that you want to reload, or choose *Reload All Config* to reload all the settings.
4. Configure the Virus Scan Interface (VSI) setting:
   a. Go to ▶ *System* ▶ *Edit Configuration Settings* ▶.
   b. Go to ▶ *System* ▶ *Env* ▶ *Virus Scan Interface Setting* ▶.
   c. Enter **`com.sap.fs.uw.cms.service.impl.VirusScanServiceForMock`** in the *Override* field for VirusScanService.
   d. Save your changes.
   e. Go to ▶ *System* ▶ *Reload Configuration Settings* ▶

A list of options appears in the tree panel.

f. Select the configuration setting type that you want to reload, or choose *Reload All Config* to reload all the settings.

3.5 Configuring the Toolkit for Content Management Systems (CMS)

To enable evidence uploads in your development environment, you need to configure settings for your CMS.

> → Tip
>
> When doing custom FS-QUO development and connecting to a new CMS Server, a certificate change may be needed on the machine hosting the runtime application.

Changing the binding type and other session parameters

The default implementation is using "BROWSER" binding type when connecting to the CMS Server. When connecting to a different CMS, that value may have to be changed. Other common values are ATOMPUB and WEBSERVICES. The value can be changed through the override of the following class and method:

Class:	`SmdRepositoryConnector`
Method:	`getSessionParams`

Code to override:	`sessionParams.put(SessionParameter.USER, userName);`
	`sessionParams.put(SessionParameter.PASSWORD, password);`
	`sessionParams.put(SessionParameter.BROWSER_URL, serverUrl);`
	`sessionParams.put(SessionParameter.BINDING_TYPE,`
	`BindingType.BROWSER.value());`
	`sessionParams.put(SessionParameter.COOKIES, "true");`

Getting the right repository on the CMS Server

The default implementation uses the "My Documents" repository of the CMS Server. The value is set in the code, and can be changed through the override of the following class and method:

Class:	`SmdRepositoryConnector`
Method:	`createSession`
Code to override:	`Repository repository = repositories.get(i); if(repository.getName().trim().equals( "My Documents"))`

Changing the upload folder within repository

The default implementation is using the "`/<CMSServerUser>` folder within the repository for file upload. This means all the uploads are stored in the folder with the same name as the CMS user account. This logic can be customized through the overrides of the following code:

Class:	`SAPMobileDocumentsService`
Method:	`uploadContent`
Code to override:	`String mobileDocUserName = SystemConfig.getInstance().getPropert y("CMSServerUser"); String fullFolderPath = "/" + mobileDocUserName + folderPath;`

Guide to main implementation classes of evidence upload and download

SAPMobileDocumentsService	Orchestrates connection to CMS Server and file up-load/download
CMSUploadProcessor	Invokes upload logic through CMSServiceFactory
CMSDownloadProcessor	Invokes download logic through CMSServiceFactory
SmdRepositoryConnector	Manages session with the CMS Server.
CMSServiceFactory	Creates an instance of CMSService through reflection.

3.6 Configuring the FS-QUO Toolkit for FS-QUO

Context

At this point, the FS-QUO development environment is only set up as the Runtime engine, but it doesn't yet have the complete setup necessary for underwriting. You still need to update the configuration settings.

If your product requires FS-PM and SAP GP-FS integration, refer to the *Integration Guide for Coverage-based Insurance Solutions* for instructions to configure integration settings in the FS-QUO Administrative Console.

3.6.1 Updating Configuration Settings

The configuration settings need to be updated as part of the FS-QUO installation process.

Context

To update the configuration settings, perform the following steps:

Procedure

1. Launch Internet Explorer and log in to the Runtime Administrative Console at the following location: `<quo-toolkit-url>/csiroot/admin/`.
   The Runtime Administrative Console will open after a short delay.
2. When prompted, log in as `<admin_user>`.
3. Choose *System* > *Edit Configuration Settings* from the menu bar.
4. Go to *Configuration* > *Application* > *AuthoritySuite* > *Env* > *Application Environment* .
   The Application Environment settings display.
   a. In the *Override* field for the `FSPMBookingServiceEnabled` row, enter **YES**.
   b. Save your changes.
5. Go to *Configuration* > *System* > *Env* > *Content Setting* .
   The Content Setting settings display.
   a. Confirm that your Application Data Model Product is present in the *Override* field for the `ApplicationDataModelProduct` row. If it is not present, enter it.
   b. Save your changes.

6. Choose *System* > *Reload Configuration Settings* from the menu bar.
7. Choose *Reload All Config*.

3.7 Copying and Deploying Insurance Product JAR Files from the Product Modeler

The FS-QUO toolkit requires insurance products to be deployed to provide meaningful contents in runtime. You will need to deploy the products to the FS-QUO toolkit.

Prerequisites

Your organization should already have a Product Modeler instance set up and insurance products imported or developed.

Procedure

1. Launch Internet Explorer and log in to the Product Modeler at the following URL: `https://<pro_designtime_app_url>/csiroot/ii/pc/`
   The Product Modeler will open after a short delay.
2. Ensure that the products selected for deployment have been published.
3. Download the product JAR files from the above location to the system where the FS-QUO toolkit is set up.
   a. Open the product  .
   b. Go to *File* > *Download Product (JAR)*.
      A dialog appears.
   c. Choose *Download*.
      A system dialog appears.
   d. Choose *Save* and specify where you want to store the JAR file.
   e. Select *Logout* at the upper-right of the page.
4. Open Eclipse using the FS-QUO toolkit workspace, and ensure that the server is started.
5. Launch Internet Explorer and log in to the Runtime Administrative Console at the following location: `<quo_toolkit_url>/csiroot/admin/`

   The Runtime Administrative Console will open after a short delay.
6. Select *Product* > *Deploy Products* from the menu bar.
7. For each insurance product JAR file to be deployed, perform the following actions:
   a. Browse to the location of the JAR file.
   b. Select the file.
   c. Choose *Upload*.

8. Verify that the product JAR files are deployed and registered successfully by validating that the same set of insurance product JAR files now exist in `<toolkit_dir>\FS-QUO-toolkit\FS-QUO-dev\target\home\ps\product`.

3.8 Installing and Configuring Standalone SAP JCo (Optional)

> **Note**
>
> SAP JCo is only releveant for FS-PM and SAP GP-FS integrations.

SAP JCo (both version 2 and 3) is required by FS-QUO to make RFC and APIF calls to FS-PM and SAP GP-FS in an integrated landscape. For an insurance product requiring integration, such as the `Sample Household Product`, you must install SAP JCo and make it available to the standalone server to write new business.

> **→ Tip**
>
> For SAP JCo 2.x, go to the *Support Packages & Patches* section of the SAP Download Center. Then go to *Archive by Alphabetical Index (A-Z)* and select *J*. Next, choose *SAP JCO* and then select *SAP JCO 2.1*.

After you have downloaded the software, install it by unzipping the file to a local directory and copying the `sapjco.jar` (version 2), `sapjco3.jar`, and `sapjco3.dll` files to the `<tomee_dir>\lib` directory. Then restart the TomEE Server from Eclipse.

Related Information

Prerequisites for Installing the FS-QUO Toolkit [page 4]

3.8.1 Configuring JCo Standalone Settings in FS-QUO

In a standalone development enviroment, you will need to define standalone JCo settings to connect to integrated SAP systems.

Prerequisites

Confirm the destination names for FS-PM and SAP GP-FS RFC connectivity that's defined in the RFC reference object in FS-PRO.

They will be referred to as `<FS-PM_dest_name>` and `<BP_dest_name>` respectively in the following steps.

For more information, see Configuring RFC Destinations in Products.

Procedure

1. Create a properties file for the FS-PM connection.
   a. Open a text editor.
   b. Add the following text to the file and add the appropriate JCo client settings in the file with connection details for the FS-PM system:

```
jco.client.client=
jco.client.user=
jco.client.sysnr=
jco.client.ashost=
jco.client.lang=
```

   where:

   client

   Specifies the SAP client number.

   Three-digit client number; preserve leading zeros if they appear in the number.

   user

   Specifies the login user ID.

   User name for logging in to the SAP system.

   Ensure that the User ID entered here has all of the required permissions in FS-PM.

   This is a technical user account. SSO/Principal Propagation is not supported.

   sysnr

   Indicates the SAP system number.

   ashost

   Identifies the SAP application server.

   Host name of a specific SAP application server.

   lang

   Specifies a login language.

   Use the SAP-specific single-character language code.

   c. Save the file as `FSPM_DEST.jcoDestination` to a local directory.
2. Create a properties file for the FS-PM connection.
   a. Open a text editor.
   b. Add the following text to the file and add the appropriate JCo client settings in the file with connection details for the SAP GP-FS system:

```
jco.client.client=
jco.client.user=
jco.client.sysnr=
jco.client.ashost=
jco.client.lang=
```

```

```

where:

client

Specifies the SAP client number.

Three-digit client number; preserve leading zeros if they appear in the number.

user

Specifies the login user ID.

User name for logging in to the SAP system.

Ensure that the User ID entered here has all of the required permissions in FS-PM.

This is a technical user account. SSO/Principal Propagation is not supported.

sysnr

Indicates the SAP system number.

ashost

Identifies the SAP application server.

Host name of a specific SAP application server.

lang

Specifies a login language.

Use the SAP-specific single-character language code.

c. Save the file as `BP_DEST.jcoDestination` to a local directory.

3. Launch Internet Explorer and go to `<quo_runtime_app_url>/csiroot/admin/`.
4. Log in to the Runtime Administrative Console as `<admin_user>`.
   The Runtime Administrative Console will open after a short delay.
5. Upload the files to the file system.

   a. Go to *System* > *Edit File System*.

   A list of folders appears in the tree, and the folders and files in the selected folder appear in the details panel.

   b. From the *Path* dropdown list, select *PA - Custom Web Root (Write)*.
   c. Expand the `resource` folder in the tree and go to the `rfcdestination` folder.
   d. Right-click the `rfcdestination` folder, and choose *Upload*.

   The *Upload Files* dialog appears.

   e. Upload both of the destination properties files.

   You can upload a maximum of eight files at one time.

> **Note**
>
> If you have the following entries already configured within *Configuration* > *System* > *Env* > *RFC Standalone Credential Setting* then you don't need to proceed with steps 8 and 9 unless you are altering the existing names:
>
> - **`FSPM_DEST_STANDALONE_PWD`**
> - **`BP_DEST_STANDALONE_PWD`**

6. Choose *System* > *Edit Configuration Settings* from the menu bar.
7. Select *Configuration* > *System* from the configuration tree.
8. Add a new configuration variable for your FS-PM connection password entry.
   a. In the editor pane, select RFC Standalone Credential Setting from the *Configuration* dropdown list.
   b. Choose the *Add Config Variable* button.
   c. Enter **`FSPM_DEST_STANDALONE_PWD`** in the *Name* field.
   d. Enter the password for `<FS-PM_dest_name>` in the *Description* field.
   e. Ensure that the *Mandatory* checkbox is deselected.
   f. Enter **`Encrypt Text`** in the *Control* field.
   g. Ensure that the *Enumeration* field is blank.
   h. Ensure that the *Default* field is blank.
   i. Save your changes.
9. Add a new configuration variable for your SAP GP-FS connection password entry.
   a. In the editor pane, select RFC Standalone Credential Setting from the *Configuration* dropdown list.
   b. Choose the *Add Config Variable* button.
   c. Enter **`BP_DEST_STANDALONE_PWD`** in the *Name* field.
   d. Enter the password for `<BP_dest_name>` in the *Description* field.
   e. Ensure that the *Mandatory* checkbox is deselected.
   f. Enter **`Encrypt Text`** in the *Control* field.
   g. Ensure that the *Enumeration* field is blank.
   h. Ensure that the *Default* field is blank.
   i. Save your changes.
10. Go to *Configration* > *System* > *Env* in the configuration tree and select *RFC Standalone Credential*.
11. Enter the password of the user for the RFC connection to the FS-PM system in the *Override* field of the `FSPM_DEST_STANDALONE_PWD` property and choose *Save*.
12. Enter the password of the user for the RFC connection to the SAP GP-FS system in the *Override* field of the `BP_DEST_STANDALONE_PWD` property and choose *Save*.
13. Select *System* > *Reload Configuration Setting* from the menu bar and choose *Reload All Config* for the new password settings to take effect.

3.8.2 Testing RFC Destinations

You need to verify that the connectivity defined in the destinations is functional.

Procedure

1. Log in to the Administrative Console.

   The Administrative Console will open after a short delay.
2. Select *System* > *Test SAP Server Connectivity*.

3. Go to the *Test JCo3 Feature* tab, select the destination and choose *Connect* to test the connection.
4. Repeat these steps for all destinations.

Results

A pop up message indicates whether the connection is valid or invalid.

3.9 Creating a User and Assigning Authorization

You will require a user for your local development environment and assign it the necessary authorizations to access the platform and the apps.

> **Note**
>
> Developers who are customizing Fiori apps should have taken an Open SAP course on Fiori application developments, or are working closely with developers who have experience with Fiori application development experience.

You will need to create a user in the Administrative Console *Role Painter* specifically for FS-QUO. This user is required to authenticate the session established between the Cloud Platform account and the on-premise FS-QUO application server.

Ensure that the user has been assigned the following roles in *Role Painter*:

- `QUO_UW_HO_LOB_PNC`
- `SAP_UW_HOUSEHOLD`
- `SAP_UW_LIFE`
- `UWFullAccess`
- `UWSUBMISSION_DATA_ENTRY`
- `UW_User_RC`
- `assignSubmission`
- `uwCaseMassReassign`
- `uwSimulate`
- `uwSubmissionListQuery`

For more information about *Role Painter*, see Product Web Services User and Authorization Management.

3.10 Setting up a Personal Development Subaccount on SAP Cloud Platform Neo

You need to create a personal development subaccount on SCP Neo, which will be used for Fiori application development.

1. Ensure that a Global account is created on SCP Neo.
2. Create an individual subaccount for each developer.
3. For each subaccount, follow the instructions on the Getting Started topic.

3.11 Setting Up Connectivity to the FS-QUO Front-End

In order to view the apps in a development environment, you will need to establish a connection to the FS-QUO front-end.

Prerequisites

You must have an SAP Cloud Platform account (either a customer account or a trial account).

You have already enabled SAP Web IDE Full-Stack in your account.

A user must have already been created in *Role Painter* specifically for FS-QUO. This user is required to authenticate the session established between the Cloud Platform account and the on-premise FS-QUO application server

Context

Procedure

1. Install the SAP Cloud Connector on the local development system.
2. Set up the Cloud Connector to connect to the SAP Cloud Platform subaccount.
   a. Start the Cloud Connector.

      For more information, see SAP Note 2485510
   b. Complete the initial configuration of Cloud Connector.

      For more information, see Initial Configuration.

> **Note**
>
> Ensure that you enter the full address of the Region instead of using the provided dropdown values. For example, enter **`int.sap.eu2.hana.ondemand.com`** instead of choosing *Europe(Frankfurt)*.

c. Select *Cloud To On-Premise* menu item for the sub-account.
d. Select the *ACCESS CONTROL* tab.
e. Add a new virtual host to the internal system mapping. This mapping will allow the exposure of the on-premise system (FS-QUO application server) to the SAP Cloud Platform subaccount. Configure it via the following properties:
   - *Back-end Type*: Select *Non-SAP System* from the dropdown list.
   - *Protocol*: Select *HTTP* from the dropdown list.
   - *Internal Host*: Enter the name of the internal host name (the host name for the on-premise system). For example, localhost or the designated IP of the server machine.
   - *Virtual Host*: Assign a name for the virtual host in the text field.
   - *Internal Port*: Enter the internal host port number (the port number of FS-QUO application server instance).
   - *Virtual Port*: Assign a name for the virtual port in the text field.
   - *Principal Type*: Select *None* from the dropdown.
   - *Host In Request Header*: Use the *Virtual Host* value.
   - Click *Check Result* in the actions menu and ensure that the result is *Reachable*.
f. Add the new accessible resource for system mapping.
   - *URL Path*: Enter the base URL path for FS-QUO (`/csiroot`).
   - *Active*: Ensure that this checkbox is selected.
   - *Access Policy*: Select the *Path and all sub-paths* radio button.
g. Verify in the *Cloud Connectors* section of the SAP Cloud Platform Cockpit that the Cloud Connector is connected and that the back-end system is exposed.

3. Create and set up an SCP destination in the SAP Cloud Platform Cockpit called `FS_QUO` with the following configuration:
   a. Select **`HTTP`** from the *Type* dropdown.
   b. Enter **`<url>/csiroot`** in the *URL* field.

      The value for `<url>` is the URL of the virtual host
   c. Select **`OnPremise`** from the *Proxy Type* dropdown.
   d. Select **`BasicAuthentication`** from the *Authentication* dropdown.
   e. Enter the user name and password of the FS-QUO user from *Role Painter*.

      Go to the *Additional Properties* tab.
   f. Enter **`TRUE`** in the *WebIDEEnabled* value field.
   g. Set the *WebIDESystem* value to **`API`**.
   h. Set the *WebIDEUsage* value to **`odata_gen`**.
   i. Save your changes.
   j. Confirm that the connections works by selecting *Check Connection* .

Related Information

Prerequisites for Installing the FS-QUO Toolkit [page 4]

3.11.1 Importing and Deploying Apps to the Fiori Launchpad

You will need to setup the FS-IPW apps in the Fiori Launchpad sandbox.

Procedure

1. Copy `FS-QUO-toolkit-<rel>.<sp>.<pl>-frontend.zip` to your local machine and extract it.
2. Go to the `Web IDE Full-Stack` service for your Neo Subaccount and ensure that the service is enabled.
3. Select *Go To service* in the *Overview* tab to open SAP Web IDE link.
4. Select *Default Workspace* on the left panel.
5. Import all of the apps, one by one, into a default workspace by performing the following steps:
   a. Go to *File* > *Import* > *File or Project* >.
   b. Browse the app ZIP file extracted from the main FS-QUO toolkit front-end ZIP file.
   c. Click *Import*.
   d. Expand the imported app, open `package.json` and add the following line in *devDependencies* `"vscode-uri": "1.0.6"` (Ref: SAP Note 2801814)
   e. Save the file.
6. Right click on the project that you want to deploy to the Fiori Launchpad and select *Deploy* > *Deploy to SAP Cloud Platform* > *Deploy a new application* >.

   If you receive the following error message `The account must have a subscription to SAP Fiori Launchpad'`, you need to enable Portal service under User Experience in Services tab to subscribe to FLP.
7. After the new applications get deployed, select *register to SAP FLP* in the new popup. The application configuration should be the same as the table below:

App Name	Semantic Object	Action	Title	Icon	Type
fs.ipw.insquote.create	InsuranceQuote	create	Create Insurance Quote	create	Static
fs.ipw.inspolicy.manage	InsurancePolicyFor-Quote	manage	Manage Issued Policies	request	Static
fs.ipw.myinsworklist	InsuranceQuote	manage	My Insurance Worklist	task	Static
fs.ipw.myuwworklist	UnderwritingCase	manage	My Underwriting Worklist	task	Static
fs.ipw.grpinsquote.create	GroupInsuranceQuote	create	Create Group Insurance Quote	create	Static
fs.ipw.myinstasks	InsuranceTask	manage	My Insurance Tasks	task	Static

App Name	Semantic Object	Action	Title	Icon	Type
fs.ipw.mstrinsquote.create	MasterInsuranceQuote	create	Create Insurance Quote From Master	create	Static
fs.ipw.mstrinspolicy.create	MasterInsurancePolicy	create	Create Master Insurance Quote	create	Static

> **Note**
>
> The Semantic Object and Action values are case-sensitive.

All apps must be registered to the same Fiori Launchpad site that you defined for your local environment.

Next Steps

If you get `401 authorization` or `fail to load SAPUI5 Component` errors while accessing the app, perform the following steps:

1. Open the Runtime Administrative Console
2. Go to *User* > *Edit Users*.
3. Search for the user that you created in Creating a User and Assigning Authorization [page 25].
4. Select the user and then choose the *Reset user's password* link on the right side.
5. Perform a hard reset to the Fiori Launchpad to load all of the apps.
6. Try to access the apps again.

3.11.2 Setting Up App-to-App Navigation

The `Create Insurance Quote from Master` app requires separate configuration to be applied for app-to-app navigation.

Prerequisites

Ensure that you have already set up connectivity to the FS-QUO front end.

Context

You only need to perform this procedure if you are developing coverage-based apps.

Procedure

1. In the *WebIDE Files* view, right-click on one of the apps and select *Project* > *Enable App to App Navigation* .

   The *App 2 App Navigation* pop-up dialog opens.
2. Select all listed apps in the *Navigate to* field and select *Enable*.
3. Once `FLPSandbox` appears in the workspace, expand the folder in the *Files* view and open `neo-app.json`.
4. Scroll to the bottom and remove all but one of the destination definitions for `FS_QUO`.
5. Change the *path* value to **`/csiroot/`** and then remove the `entryPath` entry.
6. Save the file and close it.
7. Select *FLPSandbox* in the *Files* view.
8. Select *Run* > *Run Configurations...* from the menu bar.
9. Select *App 2 App Navigation* from the menu on the left.
10. In the main tab, select the *Advanced Settings* tab. Under *Application Destinations*, select the appropriate destination name for the local FS-QUO system and select *Save and Run*.

3.12 Copying the <CSI_HOME> Artifacts to the Development Profile

After the local FS-QUO toolkit set up is complete, you should update the development profile with the artifacts from the `<CSI_HOME>` directory. Copy the following directories from `<toolkit_dir>\FS-QUO-toolkit\FS-QUO-dev\target\home` to `<toolkit_dir>\FS-QUO-toolkit\FS-QUO-dev\src\profiles\<dev_profile_name>\home`:

- `ppms\app\runtime`
- `ps\product`

The FS-QUO toolkit is now completely set up for the first time. For team development, you should check `<toolkit_dir>` into your source control system so it can be accessed by your development team. As you have already done most of the setup, each subsequent developer would only need to perform the following steps to set up their own FS-QUO development environment:

1. Installing and configuring Eclipse.
2. Installing and configuring TomEE+.
3. Installing and configuring the FS-QUO toolkit.
   For *Unzipping the FS-QUO toolkit*, replace the instruction with checking out the toolkit directory from source control.

After that, the development team can verify the integrated SCM tools within Eclipse and link each project to your source control system for updates and commits.

3.13 Building the Custom FS-QUO MTA on the XSA server

Extensions and customizations made in the FS-QUO toolkit must be built and packaged so that they can be deployed to an FS-QUO runtime environment. The following custom artifact needs to be generated for deployment: `<toolkit_dir>\FS-QUO-toolkit\FS-QUO-mta\target\FS-QUO-mta-<rel>.<sp>.<pl>.mtar`

However, instead of generating artifacts for deployment manually from the FS-QUO-toolkit, you should implement a build process that does the following:

1. Check out the FS-QUO toolkit source code from your source control system.
2. Run Maven install on the FS-QUO-toolkit Maven reactor module to generate the build artifacts.
3. Archive the build artifacts to a central location, such as a Nexus repository or in the Jenkins project, for deployment.

4 Changing Default Error Report Settings

The error report on the *Exception* screen contains a summarized description of the errors. If you want to view more details, you must override the default flag in local environments by changing the flag from Yes to No.

You can find the flag, `SuppressErrorinAS` in the following location in the Administrative Console:
Application > *Product Authority* > *Env* > *Application Environment* .

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon  : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon  : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

www.sap.com/contactsap

THE BEST RUN SAP