Extending and Testing SAP Fiori Apps

April 2014

Copyright

SAP Library document classification: PUBLIC

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Excel, Outlook, PowerPoint, Silverlight, and Visual Studio are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, z10, z/VM, z/OS, OS/390, zEnterprise, PowerVM, Power Architecture, Power Systems, POWER7, POWER6+, POWER6, POWER, PowerHA, pureScale, PowerPC, BladeCenter, System Storage, Storwize, XIV, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, AIX, Intelligent Miner, WebSphere, Tivoli, Informix, and Smarter Planet are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the United States and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are trademarks or registered trademarks of Adobe Systems Incorporated in the United States and other countries.

Oracle and Java are registered trademarks of Oracle and its affiliates.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems Inc.

HTML, XML, XHTML, and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Apple, App Store, iBooks, iPad, iPhone, iPhoto, iPod, iTunes, Multi-Touch, Objective-C, Retina, Safari, Siri, and Xcode are trademarks or registered trademarks of Apple Inc.

IOS is a registered trademark of Cisco Systems Inc.

RIM, BlackBerry, BBM, BlackBerry Curve, BlackBerry Bold, BlackBerry Pearl, BlackBerry Torch, BlackBerry Storm, BlackBerry Storm2, BlackBerry PlayBook, and BlackBerry App World are trademarks or registered trademarks of Research in Motion Limited.

Google App Engine, Google Apps, Google Checkout, Google Data API, Google Maps, Google Mobile Ads, Google Mobile Updater, Google Mobile, Google Store, Google Sync, Google Updater, Google Voice, Google Mail, Gmail, YouTube, Dalvik and Android are trademarks or registered trademarks of Google Inc.

INTERMEC is a registered trademark of Intermec Technologies Corporation.

Wi-Fi is a registered trademark of Wi-Fi Alliance.

Bluetooth is a registered trademark of Bluetooth SIG Inc.

Motorola is a registered trademark of Motorola Trademark Holdings LLC.

Computop is a registered trademark of Computop Wirtschaftsinformatik GmbH.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP BusinessObjects Explorer, StreamWork, SAP HANA, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects Software Ltd. Business Objects is an

SAP company.

Sybase and Adaptive Server, iAnywhere, Sybase 365, SQL Anywhere, and other Sybase products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Sybase Inc. Sybase is an SAP company.

Crossgate, m@gic EDDY, B2B 360°, and B2B 360° Services are registered trademarks of Crossgate AG in Germany and other countries. Crossgate is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

Contents

Extending and Testing SAP Fiori Apps

This document explains the steps required to extend an SAP Fiori app.

> *Note: This document focuses on extending the UI for transactional apps and for analytical apps that use HANA Live and XS Odata.*

To see the general SAP Fiori Extensibility documentation, see the Help Portal at help.sap.com.

The SAP Fiori Toolkit provides templates for creating SAP Fiori apps and automated extensibility options. For more information see http://scn.sap.com/docs/DOC-50112 .

SAP Fiori app extensibility or creation involves the following phases.

Phase	Description
1	Setting up the UI developer environment a. Eclipse installation b. SAP NetWeaver Gateway Productivity Accelerator (GWPA) c. UI Development Toolkit for HTML5 d. Fiori Toolkit e. Configuration See {link once available}
2	Creating a new application and testing it. See http://scn.sap.com/docs/DOC-50112.
3	Extending the original application and testing it.

This document covers extending the original app and testing it.

1 Target Audience

This document is targeted at UI developers. Some experience of working in Eclipse and in ABAP is required.

2 Extensibility: Downloading an SAP Fiori App

If you want to extend a Fiori app – rather than create a new one from scratch – you need to download an app first. Then you test the app on your local server.

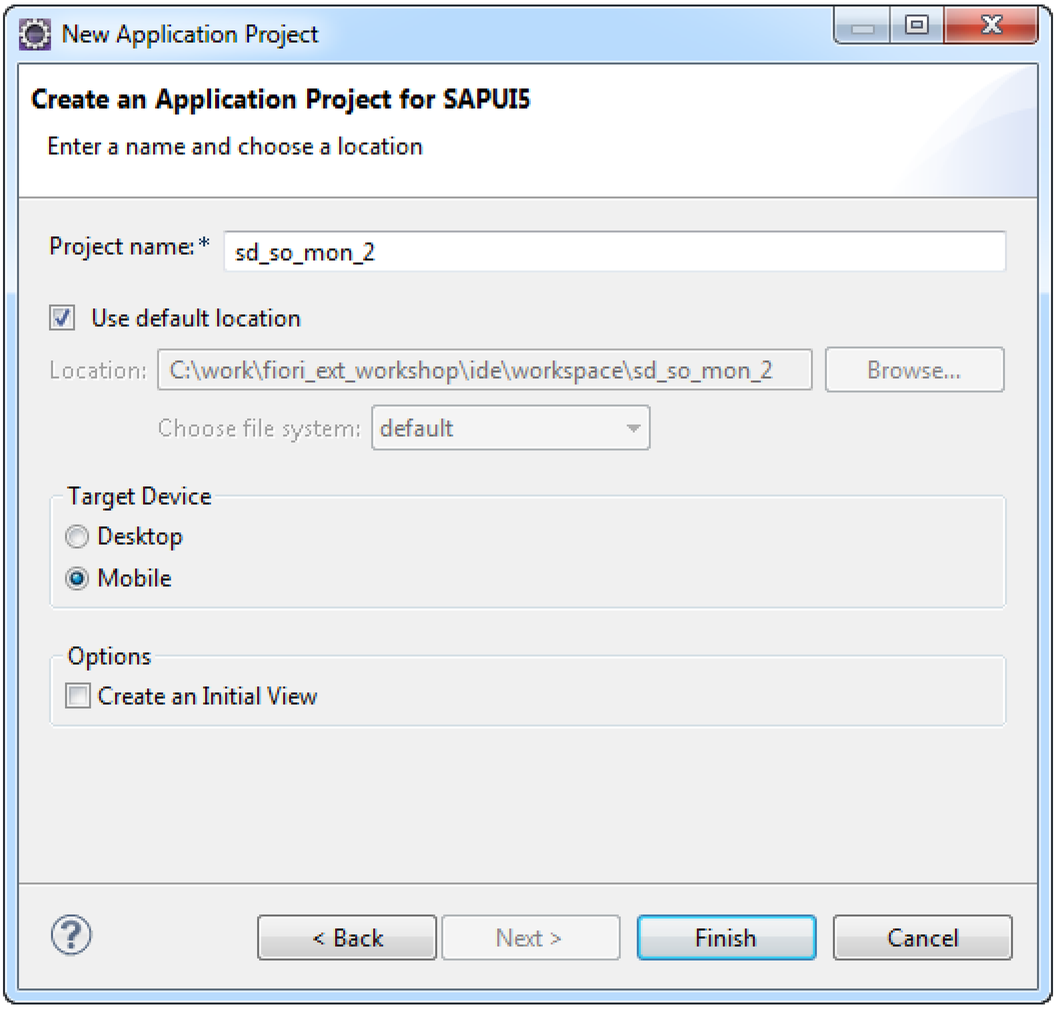
Make sure the ABAP team provider is setup before continuing. You can do this using *Eclipse*. For more information, see **https://tools.hana.ondemand.com/#abap**.

2.1 Download SAP-original UI5 App from ABAP Repository using SAPUI5 ABAP Repository Team Provider

You can download the original app in five steps. They are as follows:

Step	Description
1	Create a new UI5 Application Project in Eclipse
2	Share the project using the ABAP Repository Team Provider
3	Download all files from the ABAP Repository
4	Select files and folders to move to the WebContent folder
5	Unshare the project from the ABAP Repository

Step 1: Create a new UI5 Application Project in Eclipse



New Application Project

Create an Application Project for SAPUI5

Enter a name and choose a location

Project name:* sd_so_mon_2

☑ Use default location

Location: C:\work\fiori_ext_workshop\ide\workspace\sd_so_mon_2

Choose file system: default

Target Device

○ Desktop

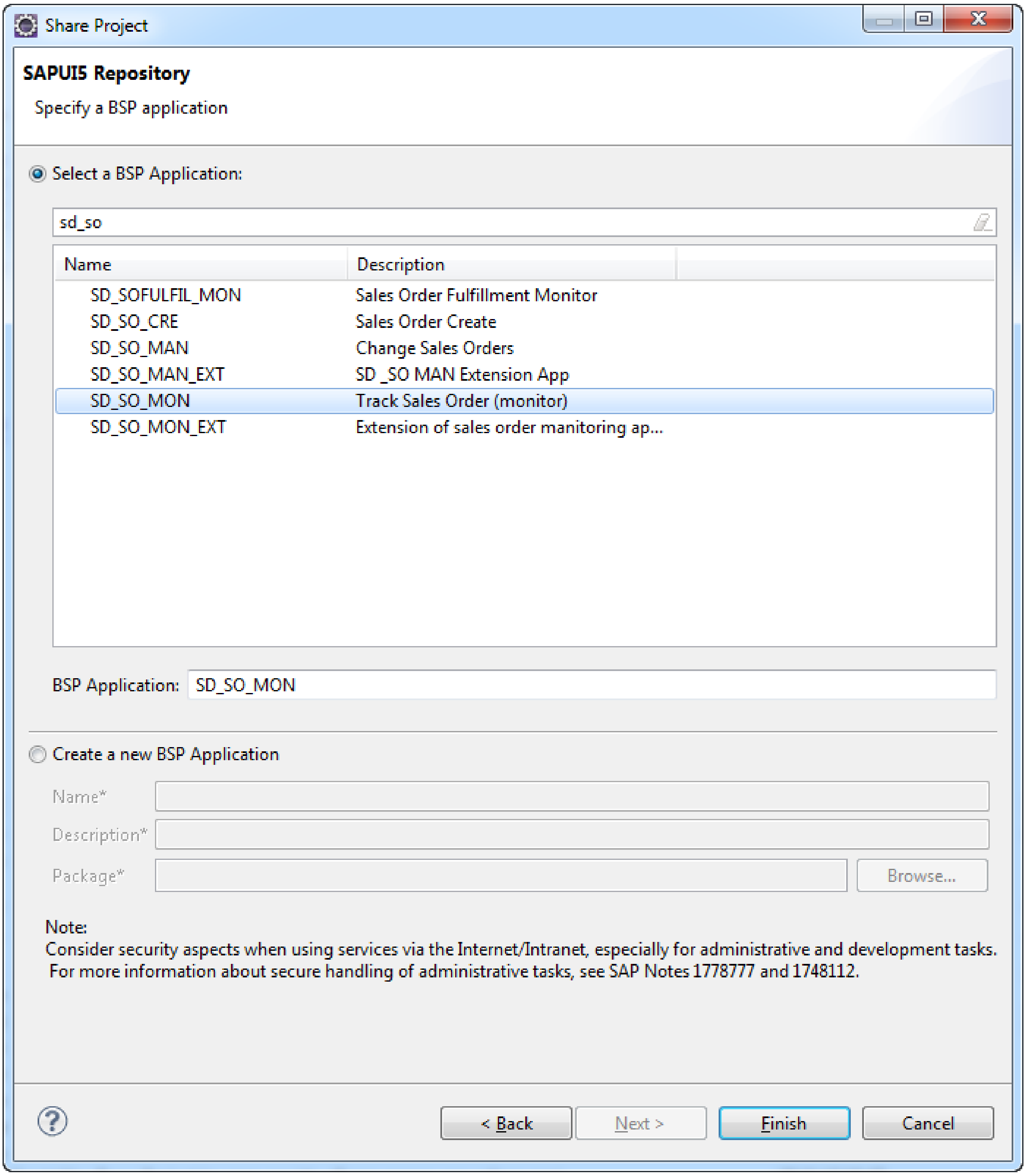
● Mobile

Options

□ Create an Initial View

Step 2: Share the project using the ABAP Repository Team Provider

1. Choose *Team* → *Share Project* on the project root node.
2. Enter the system connection. (Prerequisite: connection entry exists in SAP Logon Pad)
3. Select *BSP application* from the list.
4. Choose *Finish*.



SAPUI5 Repository

Specify a BSP application

● Select a BSP Application:

sd_so

Name	Description
SD_SOFULFIL_MON	Sales Order Fulfillment Monitor
SD_SO_CRE	Sales Order Create
SD_SO_MAN	Change Sales Orders
SD_SO_MAN_EXT	SD _SO MAN Extension App
SD_SO_MON	Track Sales Order (monitor)
SD_SO_MON_EXT	Extension of sales order monitoring ap...

BSP Application: SD_SO_MON

○ Create a new BSP Application

Name*

Description*

Package*

Note:
Consider security aspects when using services via the Internet/Intranet, especially for administrative and development tasks.
For more information about secure handling of administrative tasks, see SAP Notes 1778777 and 1748112.

Step 3: Download all files from the ABAP Repository

1. Select the project root node and choose *Team* → *Retrieve* from the context menu.
2. Choose *Finish*.

Step 4: Select files to move to the Web Content folder

From all the content downloaded from the ABAP Repository, select all visible files and folders, except the *build* folder, and move them to the WebContent folder.

> *Note: Do not move the build folder, nor any files beginning with ".". However, these files are hidden by default, so you should not move them accidentally.*

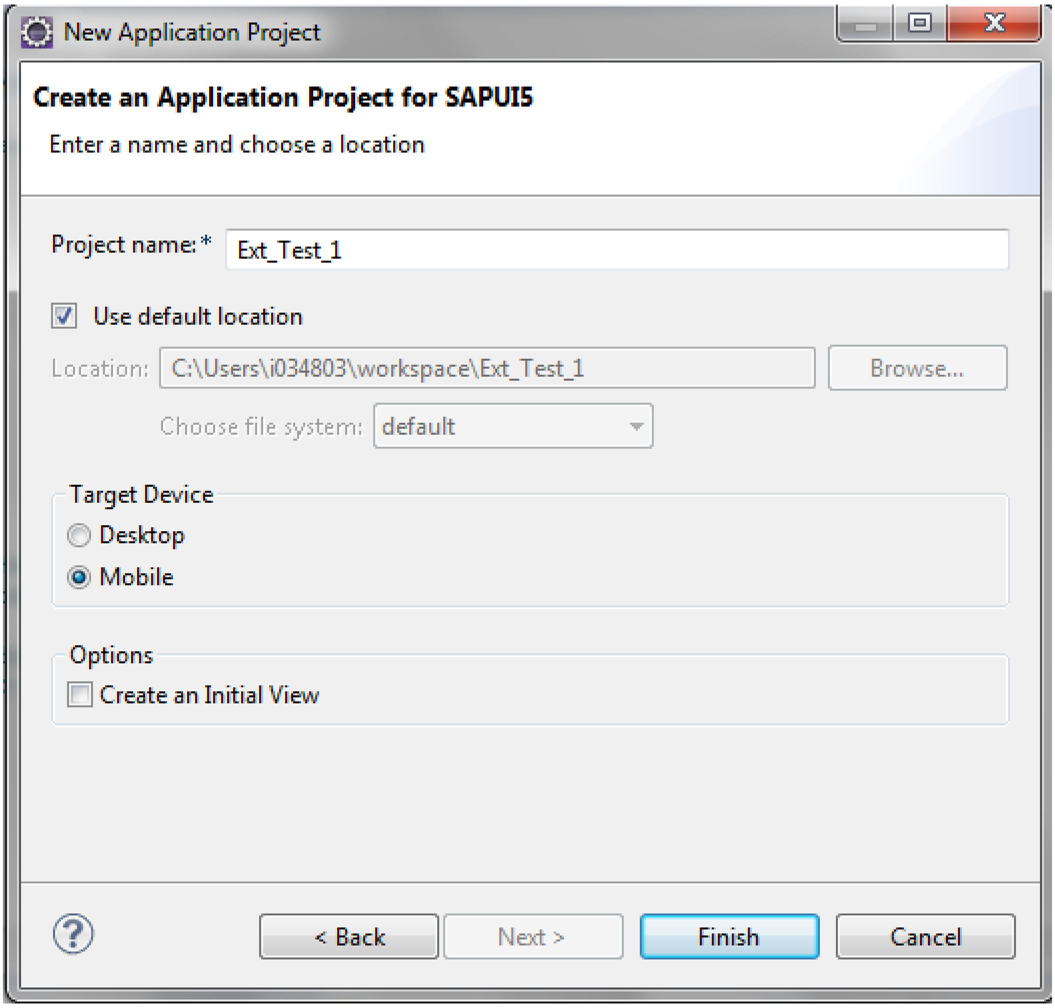
Step 5: Unshare the project from the ABAP Repository

In order to avoid unwanted modifications to the original application, unshare the project from the ABAP Repository by choosing *Team* → *Unshare Project* on the context menu of the project root node.

3 Extending the SAP Fiori App

The first step is to create new extension project in Eclipse. Select *File* → *New* → *Other* → *Application Project for SAP UI5*.

Add a project name, specify the location, select the *Mobile* check box, and clear the *Create an Initial View* checkbox.



Create an Application Project for SAPUI5

Enter a name and choose a location

Project name:* Ext_Test_1

☑ Use default location

Location: C:\Users\i034803\workspace\Ext_Test_1

Choose file system: default

Target Device

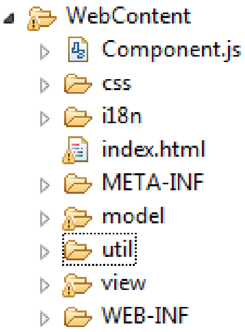
○ Desktop

● Mobile

Options

□ Create an Initial View

Now you create a view, model, and 'i18n' folder. Then create a Component.js file in the *WebContent* folder.



- WebContent
  - Component.js
  - css
  - i18n
  - index.html
  - META-INF
  - model
  - util
  - view
  - WEB-INF

3.1 Setting up Component.js

The *Component.js* file extends the SAP-delivered standard application component as shown below. You can copy this code and change it where necessary, thereby applying it to the app you want to extend. You can find the component name in the *Component.dbg.js* file of the SAP-delivered app.

Example

Here is an example showing the component name in the file.

Note *In Eclipse, the BSP application name/project name of the SAP-delivered standard application is ZF2UIAPPREF002.*

```
jQuery.sap.declare("i2d.qm.qualityissue.confirm.apprefExt.Component");

// use the load function for getting the optimized preload file if present
sap.ui.component.load({
                name: " i2d.qm.qualityissue.confirm.appref",
                url: jQuery.sap.getModulePath("i2d.qm.qualityissue.confirm.apprefExt") +
"/../ZF2UIAPPREF002"
                // we use a URL relative to our own component; might be different if
                // extension app is deployed with customer namespace
});

i2d.qm.qualityissue.confirm.appref.Component.extend("i2d.qm.qualityissue.confirm.apprefExt.
Component",
{
         metadata:{
                 /*meta data details like configuration, customizing, and routing data*/
         }
});
```

Note *The namespace for the custom application might differ from the standard SAP application. You may need to adjust this path before uploading the custom application to the ABAP system.*

Note *Ensure that the project name of the standard SAP application should be the same as the BSP application service name on the server.*

The SAP UI5 demo kit shows how to add more extensibility options.

4 Testing the Extended Application on the Local Tomcat Sandbox

Once you have the Tomcat server set up, you can test the new application locally.

See chapter 1.14.6.1 *Local Sandbox Environment for the Launchpad* in the *Developer Guide for UI Addon* at http://help.sap.com/nw-uiaddon#section3 → *Developer's Guide*.

Uploading the New Application to the SAPUI5 Repository

For information on how to upload the new application to the SAPUI5 Repository, see the section *Sharing the SAPUI5 Application Project with the SAPUI5 Repository* in the *Developer Guide for UI,* chapter 1.1.3.18.1, *Using Eclipse to Synchronize*, at http://help.sap.com/nw-uiaddon#section3 → *Developer's Guide*.

> *Note: The SAP ABAP Server requires windows-only line delimiters. If the app was created on a Mac, make sure the line delimiters are windows ones. You can do this in Eclipse by choosing* File → Convert Line Delimiters To…

Once the extended app has been uploaded to the SAP system, it can be tested.

On the Fiori launchpad sandbox, configure both the standard application and the extended application so that you can run both the applications from the Fiori launchpad sandbox.

5 Testing the Application on the ABAP System

The next step is to test integration in the shell. You need to release the transport request and then test it on the ABAP system.

For information on how to do this, see chapter 1.1.1.5, *Test Your SAPUI5 Application on an ABAP Server* in the *UI AddOn Developer Guide for User Interface Add-On* at http://help.sap.com/nw-uiaddon#section3 → *Developer's Guide*.

6 Configuring the Application on the Fiori Launchpad

The final stage is to configure the app on the SAP Fiori Launchpad.

> *Note: This is often the task of the System Administrator rather than the developer.*

For more information, see SAP Library for User Interface Add-On 1.0 for SAP NetWeaver on SAP Help Portal at **http://help.sap.com/nw-uiaddon**. In SAP Library, choose *Administration Guide* → *Content Administration* → *SAP Fiori Launchpad* → *Setting Up the Launchpad and Launchpad Designer*.

You have to change the navigation targets, then create a semantic object.
For detailed steps on how to do this, see Customizing Navigation Targets and Creating Custom Semantic Objects.