



## How to Extend Fiori Applications

# TABLE OF CONTENTS

1.	Document Overview .....	3
1.1.	Purpose .....	3
1.2.	Out of Scope.....	3
1.3.	Technical Background.....	3
1.4.	UI Extensibility .....	3
2.	Technical Requirements.....	4
2.1.	Prerequisites .....	4
2.1.1.	Setting up Eclipse and UI5 Environment .....	4
2.1.2.	Using this document .....	4
3.	Step-by-Step Guide .....	4
3.1.	Output Preview Before and After Change .....	4
3.2.	Finding the corresponding BSP application of Fiori App .....	7
3.3.	Downloading the Source Code of the <i>Sales Order Tracking</i> app .....	12
3.4.	Importing the SAPUI5 Application to Eclipse. ....	14
3.5.	Deciding the Application Name .....	20
3.6.	Gateway Extensibility.....	22
3.7.	Creating a New Enhancement SAPUI5 Project .....	36
3.8.	Testing the Application Locally before Uploading .....	43
3.9.	Upload the Application into the SAP System.....	54
3.10.	Configuring the New Application in the SAP Fiori Launchpad. ....	59

# 1. Document Overview

## 1.1. Purpose

This document describes how to extend an existing SAP Fiori app (*SAP Fiori principal apps for SAP ERP* and *SAP Fiori principal apps for SAP SRM*). The example scenario involves adding a “payment terms information” field to the SAP Fiori principal app, Sales Order tracking. This scenario outlines one of the strengths of the SAP Fiori solution: how to perform a modification-free extension to an SAP Fiori app.

This document describes OData and UI extensibility for SAP Fiori principal applications (built during wave 2 or later).

## 1.2. Out of Scope

This document does not provide details to build a custom application from scratch.

## 1.3. Technical Background

SAP Fiori is a collection of apps that look not only beautiful but are also extremely easy to use. The apps run across different devices with the same user experience. With the new SAPUI5, SAP Fiori apps run on desktop, tablet or smartphone and all UI5-compatible browsers, like Safari on iOS, IE9 and higher, or Chrome.

SAP Fiori provides a role-based, homogenous experience across a variety of devices. SAP has put in high effort to mobilize workflows and productivity applications for employees, executives, and managers who need a consolidated, intuitive, and easy-to-use interface based on SAPUI5 (HTML5).

## 1.4. UI Extensibility

The extensibility concept provides a very high flexibility of extensions. In particular, the UI Extensibility concept allows a customer to make modification-free extensions for certain extensibility use cases. SAP has enabled extension points in the code of SAP Fiori apps. To find the extension points for each app, consult the app-specific documentation.

You can use the following properties of the *Component.js* file to extend the app:

- **sap.ui.viewExtensions**  
This provides custom view content at a specified extension point in the standard, SAP-delivered application. Simply, you can extend the application with new fields or edited images at extension points.
- **sap.ui.viewModifications**  
This is used for overriding certain control properties in the standard SAP-delivered application. You can use it to hide or show any property of the existing element; for example, changing the width or height of an element.
- **sap.ui.viewReplacements**  
This is used for replacing a view in an SAP-delivered application with a custom view. You should use this property if there is no extension point in a particular place/view, or if you need a major change in the application.
- **sap.ui.controllerExtensions**  
This property is used for replacing a controller in an SAP-delivered application with a custom controller. Controllers contain application business logic. Therefore, if there is change needed in the business logic, you have to use this property.

**Note:** This document only covers how to use the **sap.ui.viewExtensions** and **sap.ui.viewModifications** properties.

## 2. Technical Requirements

### 2.1. Prerequisites

#### 2.1.1. Setting up Eclipse and UI5 Environment

For details on this, refer to <http://scn.sap.com/docs/DOC-52436>.

#### 2.1.2. Using this document

All the objects mentioned in this document are fictitious. If you are following this document and creating objects, please change the name for each new object. We recommend using the customer namespace as a prefix and then adding some two-digit number to all objects (same number for all objects) as shown below.

##### Example

New application name: Z\_SD\_SO\_MON as ZS\_D\_SO\_MON\_XX (where XX is 01 or 02 or 11, or 22, etc.)

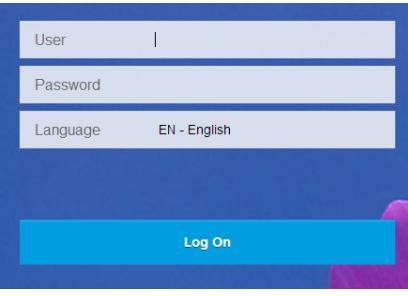
Gateway service: ZSRA018\_SO\_TRACKING\_SRV as ZSRA018\_SO\_TRACKING\_SRV\_XX

If you are keeping the different name, change the object name accordingly in the source code and the configuration. Otherwise you may end up with an error.

## 3. Step-by-Step Guide

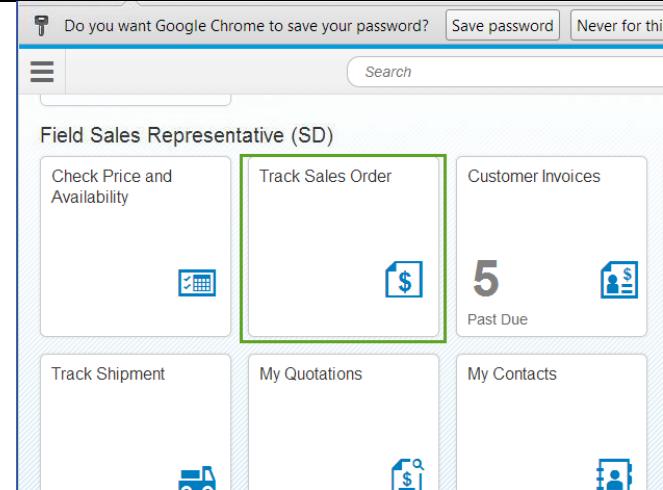
### 3.1. Output Preview Before and After Change

In this scenario we are going to display the payment term information in the header section using the property **sap.ui.viewExtensions**. We are also going to hide material information using the property **sap.ui.viewModifications**.

What to do	What to Say
<p>3.1.1. Login to Fiori Launchpad</p> <p><b>Fiori Launchpad url</b></p> <p><code>http://&lt;your_ABAP_URL&gt;/sap/bc/ui5_ui5/ui2/ushell/shells/abap/FioriLaunchpad.html</code></p> <p>Replace &lt; your_ABAP_URL&gt; with <code>http://&lt;ABAP server&gt;:&lt;port&gt;</code></p>	

- 3.1.2. Scroll down the page until you find the *Track Sales Order* application.

Click the application to open it.



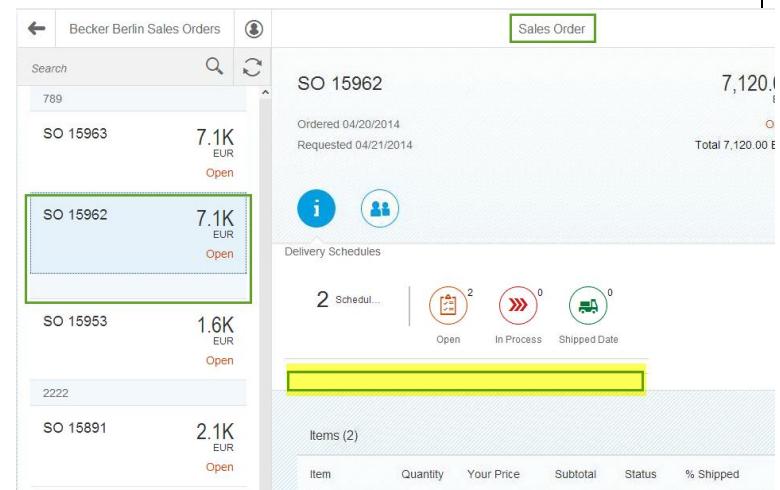
- 3.1.3. Select any sales order in the left side navigation bar.

We can see the sales order header and other information on the right side.

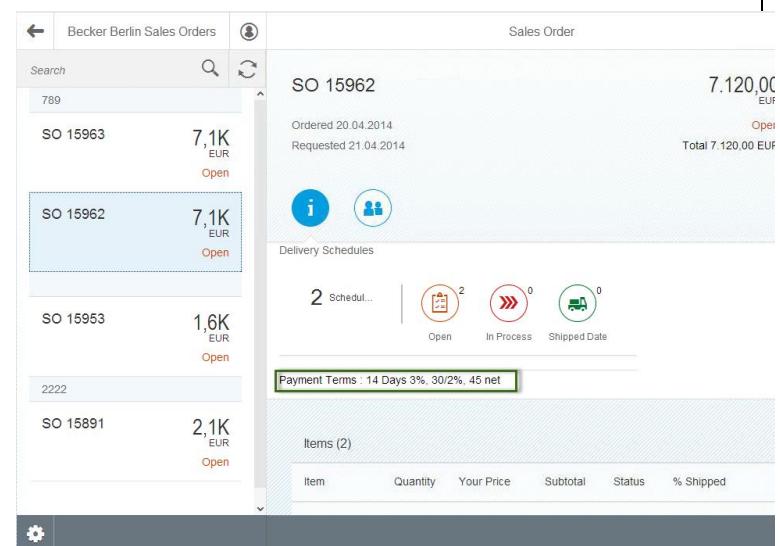
This document describes how to display the payment terms in the place highlighted in yellow

The next screenshot shows how the application looks after the change.

**Application Screenshot before change**

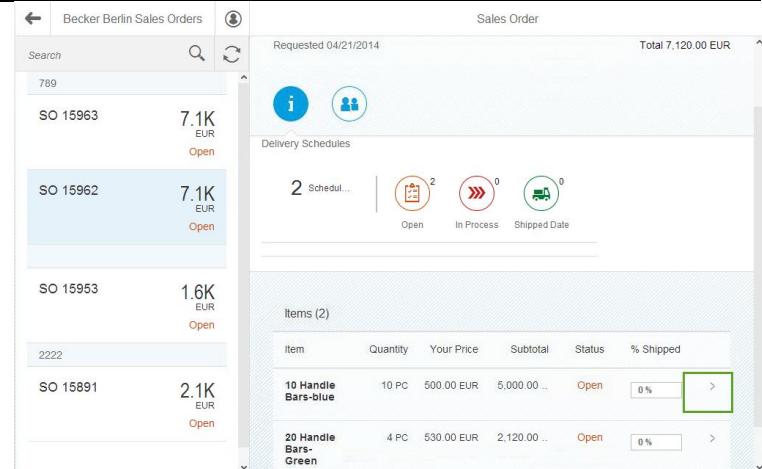


**Application Screenshot After change**



- 3.1.4. We are also going to hide a section in the item detail page, and see how the app looks after the change.

Click on the small arrow mark in the item list, as shown in the screenshot

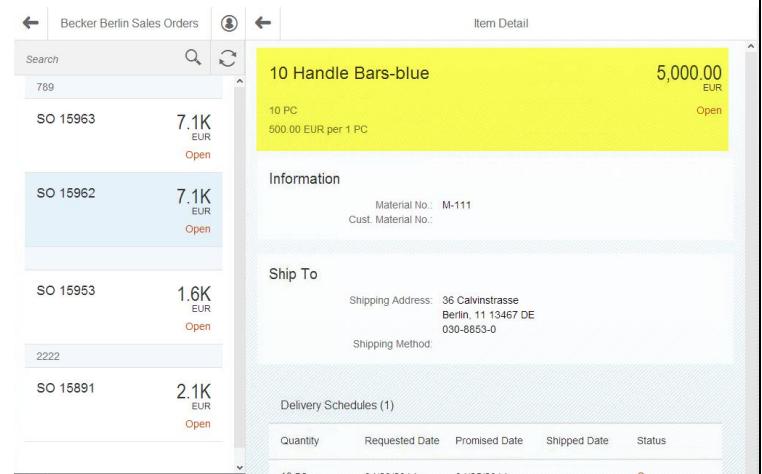


- 3.1.5. This screenshot shows the item details.

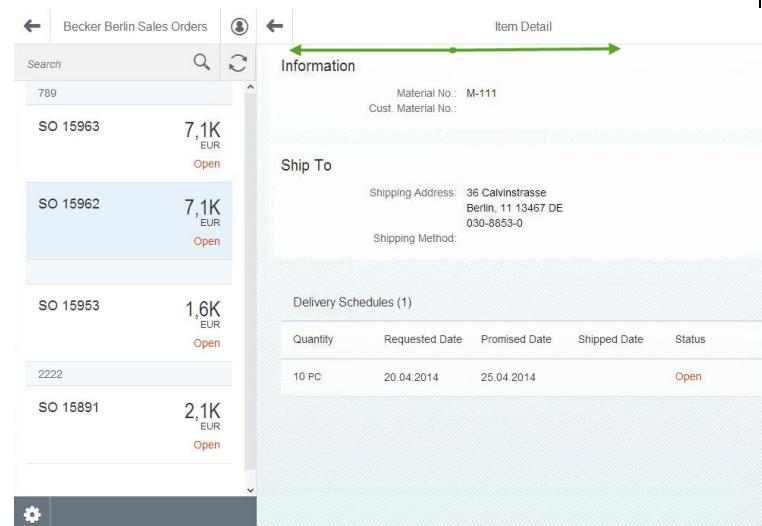
This document shows how to hide the highlighted section in the application.

The screenshots also shows how the application looks before and after the changes.

**Application screenshot before change**

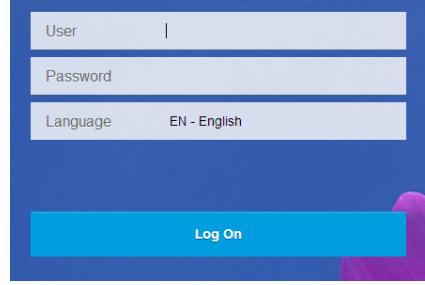
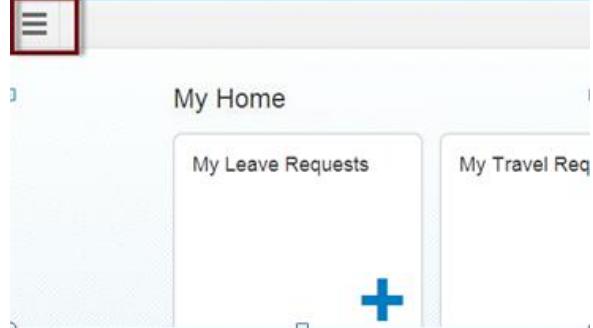
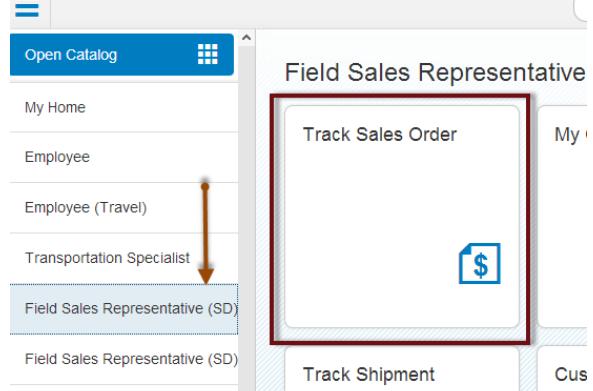
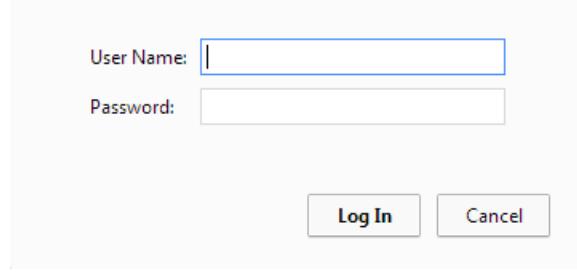


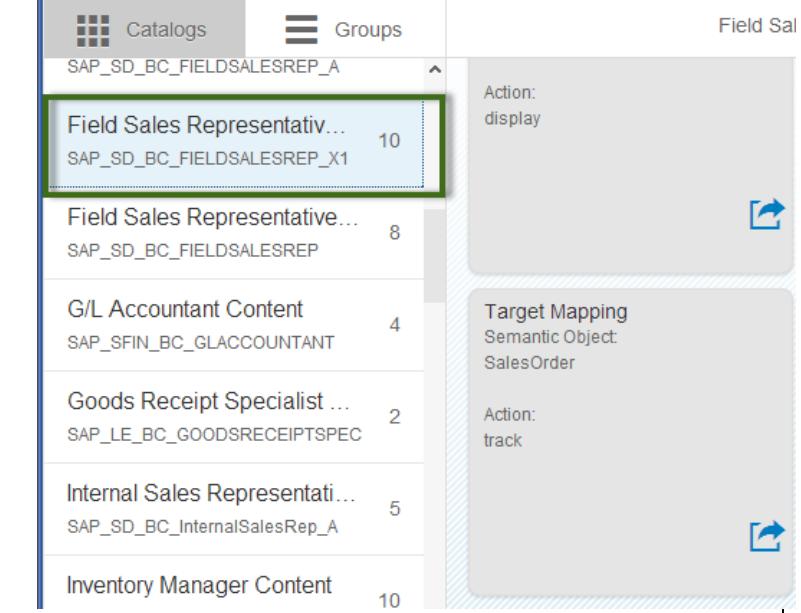
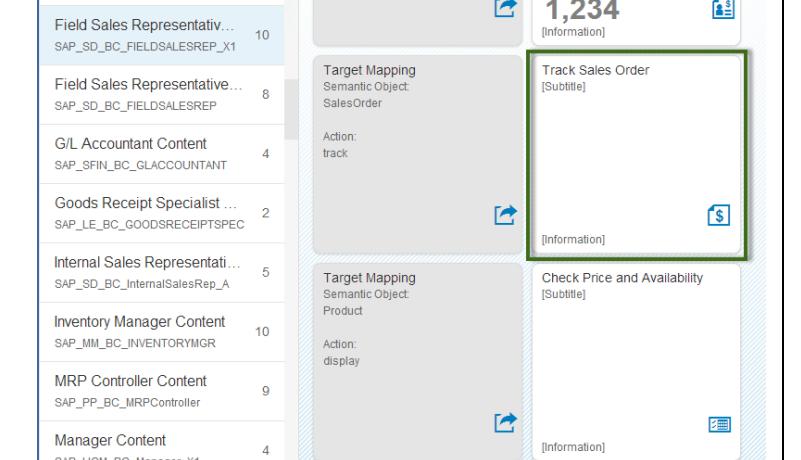
**Application Screenshot after change**



### 3.2. Finding the corresponding BSP application of Fiori App

First we need to find out the corresponding standard BSP application name of the standard SAP Fiori app to enhance the application.

What to do	What to Say
<p>3.2.1. Login to Fiori Launchpad</p> <p><b>Fiori Launchpad url</b></p> <p><code>http://&lt;your_ABAP_URL&gt;/sap/bc/ui5_ui5/ui2/ushell/shells/abap/FioriLaunchpad.html</code></p> <p>Replace &lt; your_ABAP_URL&gt; with <code>http://&lt;ABAP server&gt;:&lt;port&gt;</code></p>	
<p>3.2.2. Click the <i>Menu</i> icon.</p>	
<p>3.2.3. The <i>Track Sales Order</i> application is found in the <i>Field Sales Representative (SD)</i>. Take note of the catalog name.</p>	
<p>3.2.4. Login to the <i>Fiori Config</i> page.</p> <p><b>Fiori Config page url</b></p> <p><code>http://&lt;your_ABAP_URL&gt;/sap/bc/ui5_ui5/sap/arsvc_upb_admn/main.html?scope=CUST</code></p> <p>Replace &lt; your_ABAP_URL&gt; with <code>http://&lt;ABAPserver&gt;:&lt;port&gt;</code></p>	

<p><b>3.2.5.</b> We need to find the application in the respective catalog.</p> <p>Select <i>Field Sales Representative (SD)</i> in the catalog. (Note the catalog name as mentioned in point 3.2.3).</p> <p><b>Note:</b> If multiple catalogs exist with the same name, we need to search in all catalogs until we find our application.</p>	 <p>Action: display</p>
<p><b>3.2.6.</b> Click the <i>Track Sales Order</i> application. (That is, the same name of the application you want to enhance).</p>	 <p>Action: track</p>

3.2.7. Take note of the *Semantic Object* and the *Action* values.

Configure: 'Track Sales Order'

**General**

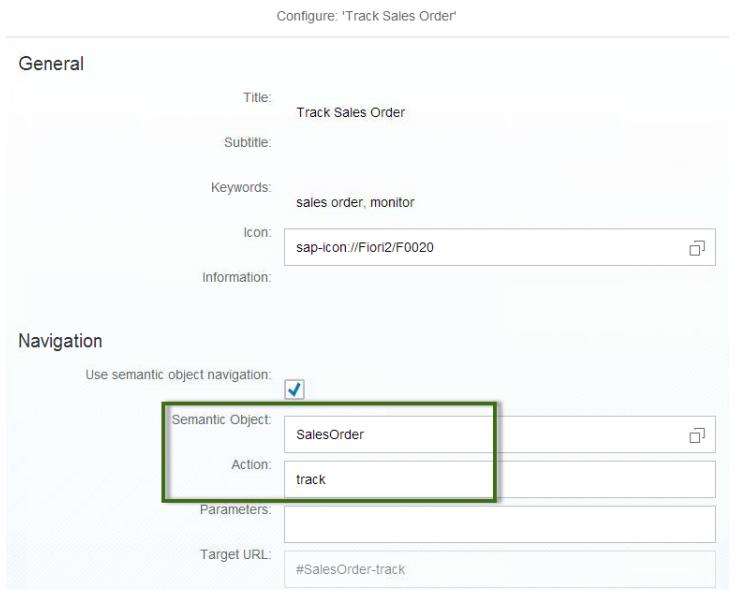
Title: Track Sales Order  
Subtitle:  
Keywords: sales order, monitor  
Icon: sap-icon://Fiori2/F0020  
Information:

**Navigation**

Use semantic object navigation:

Semantic Object: SalesOrder  
Action: track

Parameters:  
Target URL: #SalesOrder-track

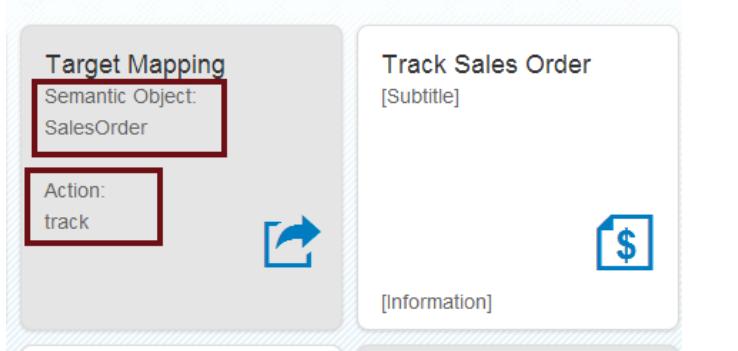


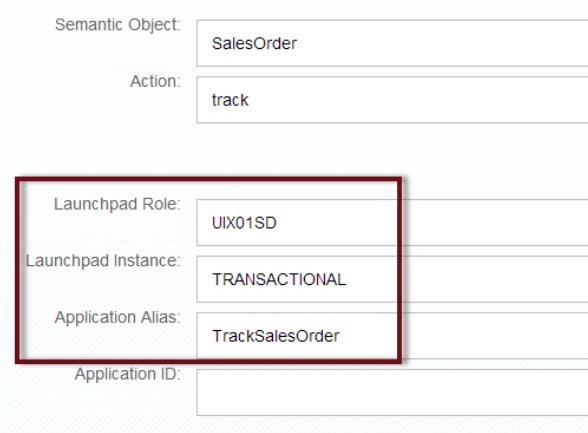
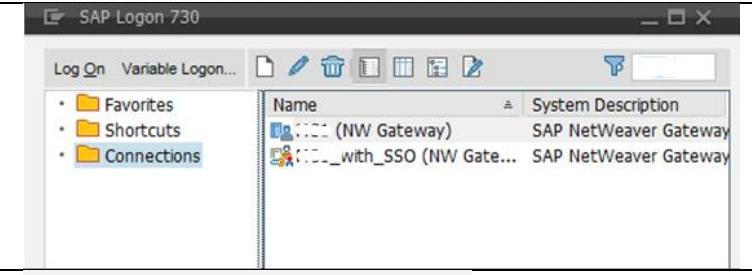
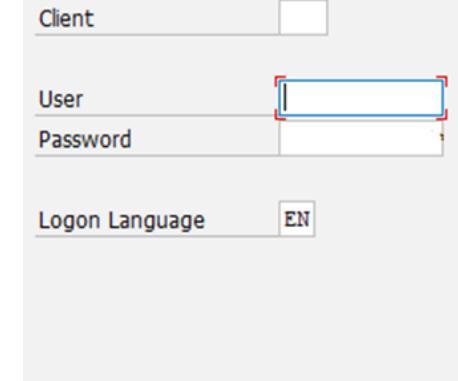
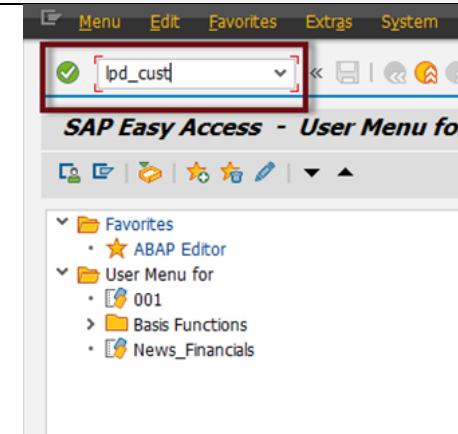
3.2.8. Click the *Back* button.



3.2.9. Click *Target Mapping*.

Note the **sales order** semantic object and the **track** action (same as we noted the semantic object and the action in point 3.2.7).

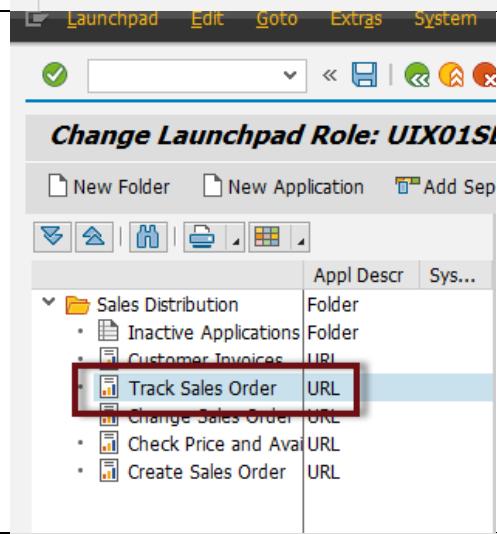


<p>3.2.10. Note the values of <i>Launchpad Role</i>, <i>Launchpad Instance</i>, and <i>Application Alias</i>.</p>	 <p>Semantic Object: SalesOrder</p> <p>Action: track</p> <p>Launchpad Role: UIX01SD</p> <p>Launchpad Instance: TRANSACTIONAL</p> <p>Application Alias: TrackSalesOrder</p>						
<p>3.2.11. Open the Gateway system.</p>	 <p>SAP Logon 730</p> <p>Log On Variable Logon...</p> <ul style="list-style-type: none"> <li>Favorites</li> <li>Shortcuts</li> <li>Connections</li> </ul> <table border="1"> <thead> <tr> <th>Name</th> <th>System Description</th> </tr> </thead> <tbody> <tr> <td>(NW Gateway)</td> <td>SAP NetWeaver Gateway</td> </tr> <tr> <td>_with_SSO (NW Gate...</td> <td>SAP NetWeaver Gateway</td> </tr> </tbody> </table>	Name	System Description	(NW Gateway)	SAP NetWeaver Gateway	_with_SSO (NW Gate...	SAP NetWeaver Gateway
Name	System Description						
(NW Gateway)	SAP NetWeaver Gateway						
_with_SSO (NW Gate...	SAP NetWeaver Gateway						
<p>3.2.12. Login with Gateway user and password.</p>	 <p>Client</p> <p>User</p> <p>Password</p> <p>Logon Language EN</p>						
<p>3.2.13. Run transaction LPD_CUST.</p>	 <p>lpd_cust</p> <p>SAP Easy Access - User Menu for</p> <ul style="list-style-type: none"> <li>Favorites             <ul style="list-style-type: none"> <li>ABAP Editor</li> </ul> </li> <li>User Menu for             <ul style="list-style-type: none"> <li>001</li> <li>Basis Functions</li> <li>News_Financials</li> </ul> </li> </ul>						

3.2.14. Double-click the respective role and instance (as we noted in point 3.2.10).

Overview of Launchpads		
UIX01MM	TRANSACTIONAL	MM Transactional Apps (Wave 1)
UIX01SD	TRANSACTIONAL	Sales Distribution
UIX01SRM	TRANSACTIONAL	SRM Wave1 Transactional Application
UIX01SRM_D	TRANSACTIONAL	SRM Wave 1 Transactional Application
UIX01TRV	TRANSACTIONAL	Travel request approval
ZUICRM001	TRANSACTIONAL	Custom CRM Fiori Application
ZUIERP001	TRANSACTIONAL	Custom Enhanced Fiori Applications

3.2.15. Select *Track Sales Order* on the left side navigation menu.

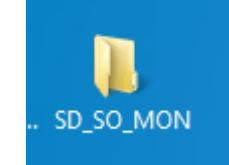
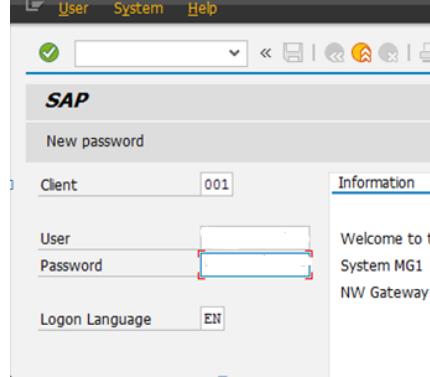
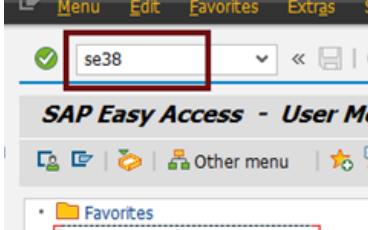
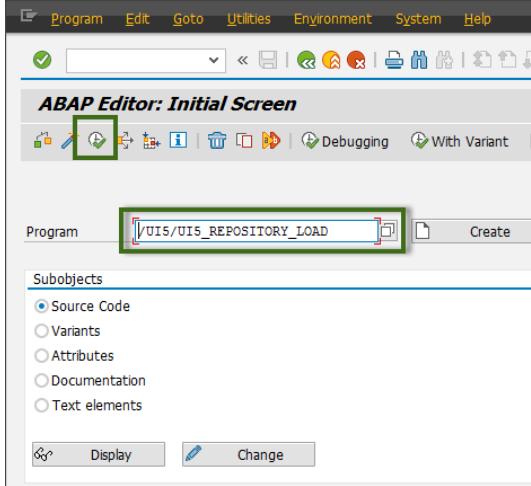


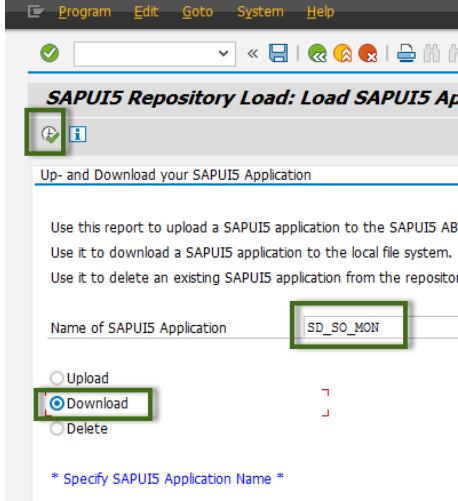
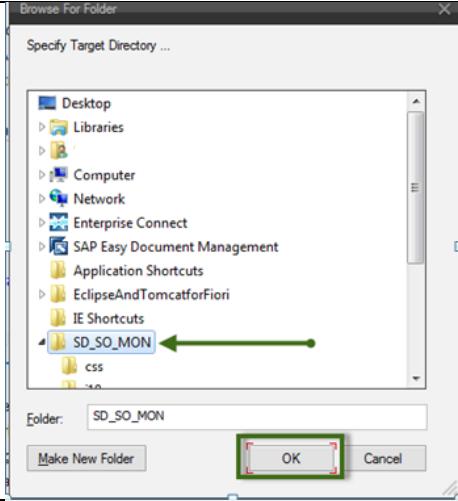
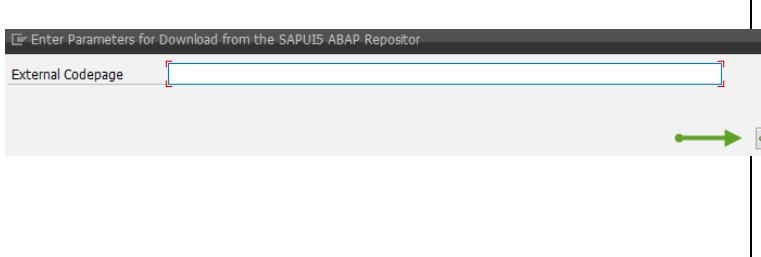
3.2.16. The *Application Alias* in the screenshot should be same as we noted in point 3.2.10.

Note the highlighted text **sd\_so\_mon** in the URL parameter. **sd\_so\_mon** is the BSP application name of *Track Sales Order*.

Link Text	Track Sales Order
Description	
Application Type	URL
URL	/sap/bc/ui5_ui5/sap/sd_so_mon
System	System Alias
<input type="checkbox"/> Hide Advanced (Optional) Parameters <b>Advanced Parameters (Optional)</b>	
<input type="checkbox"/> Application - Deactivation by User Application Cannot be Removed from Launchpad	
<b>Application-Related Parameters</b>	
Application Alias	TrackSalesOrder
Target App. Parameters	
Suspend/Resume	

### 3.3. Downloading the Source Code of the Sales Order Tracking app

What to do	What to Say
<p>3.3.1. Create a folder called <i>SD_SO_MON</i> on the Desktop.</p> <p><b>Note:</b> Do not append any number to the folder name.</p>	
<p>3.3.2. Open the SAP Gateway system from SAP Logon Pad.</p> <p>Login with username and password.</p>	
<p>3.3.3. Run transaction SE38.</p>	
<p>3.3.4. Enter the program name /UI5/UI5_REPOSITORY_LOAD and click the Execute icon.</p>	

<p>3.3.5. Enter the name of the SAPUI5 Application SD_SO_MON (refer to point 3.2.16 for the BSP application name). Select the <i>Download</i> option. Click the <i>Execute</i> icon.</p>	
<p>3.3.6. Select the SD_SO_MON folder created in the desktop.  Click <i>OK</i>.</p>	
<p>3.3.7. Scroll down to the bottom of the output page and click where it reads <i>Click here to Download</i>.</p>	 <pre> Download File      : view/S4-dbg.controller.js Download File      : view/S4.controller.js ( Text ) Download File      : view/S4.view.app.xml ( Text ) Download File      : view/S4.view.xml ( Text ) Download File      : view/S4EditDialog.fragment. Download File      : view/S4SubmitDialog.fragment. Download File      : view/SODeliverySchedules-db Download File      : view/SODeliverySchedules.co Download File      : view/SODeliverySchedules.vi  * Items to be ignored have been determined from the * File '.Ui5RepositoryBinaryFiles' has been considered * File '.Ui5RepositoryTextFiles' has been considered </pre>
<p>3.3.8. Leave the <i>External Codepage</i> field blank and click the green tick mark.</p>	

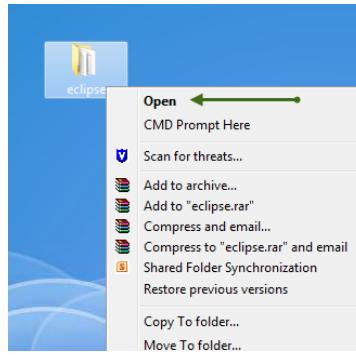
3.3.9. Once the application is downloaded, you should see this message displayed, as shown in the screenshot.

> Additional text files may be specified via '.l  
> Additional binary files may be specified via  
> Each line in the files above specifies a file or  
Either in form a sub string or as a regular expression  
Example: '^.\*[/\\\\]build(\\/[\\\\].\*)?\\\$' starts  
3.679 bytes transferred

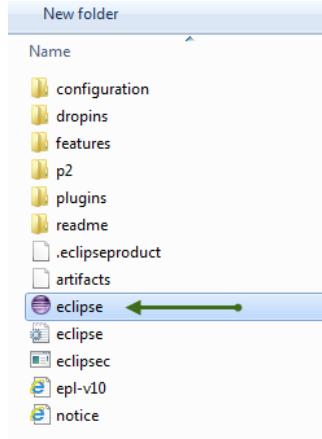
### 3.4. Importing the SAPUI5 Application to Eclipse.

In order to modify an SAP Fiori app, it is recommended to do so with an Eclipse installation that has the SAPUI5 Application Development feature and the Fiori toolkit installed. (We have already copied Eclipse with SAP Fiori toolkit installed to the desktop). This simplifies the development process.

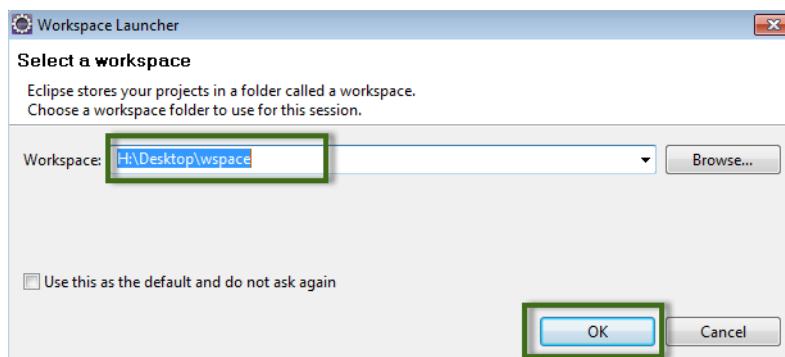
First we need to import the standard SAP Fiori application into Eclipse. To do this, open the *Eclipse* folder.



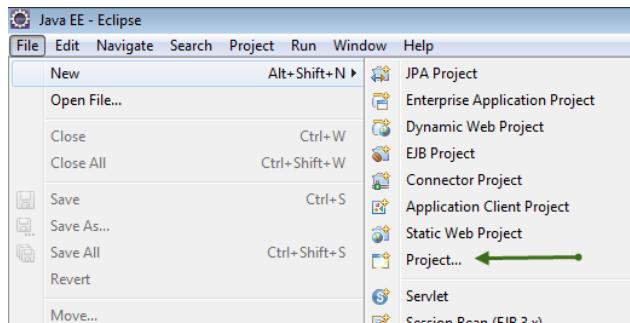
3.4.1. Open and run *eclipse.exe*.



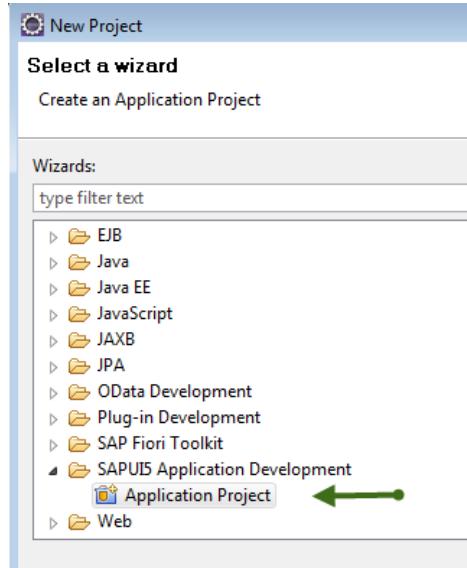
3.4.2. Enter a workspace name and click *OK*.

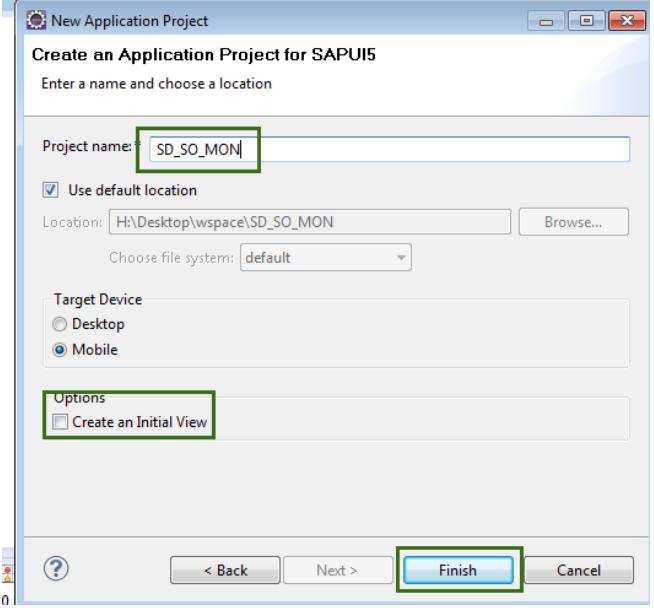
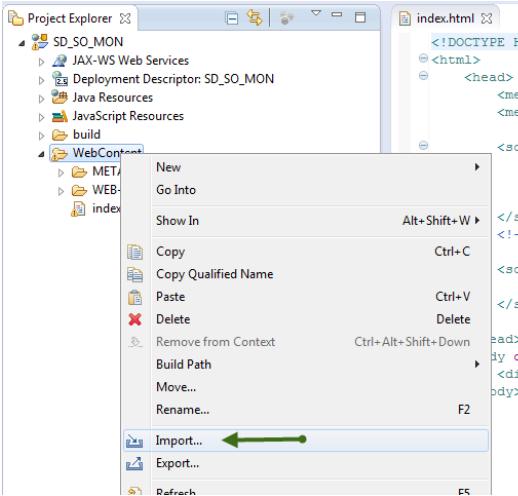
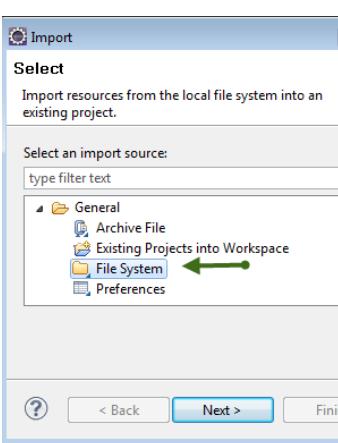


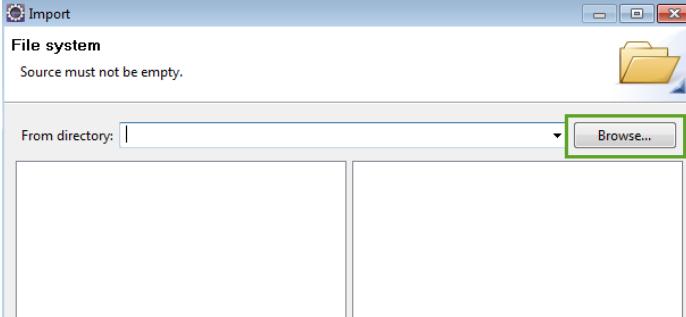
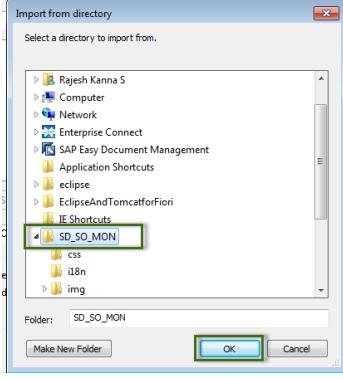
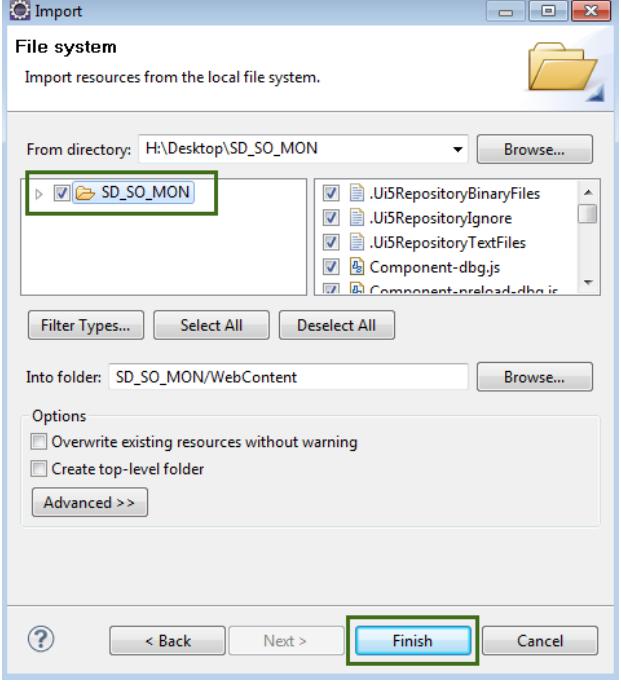
3.4.3. Select *File* → *New* → *Project*.



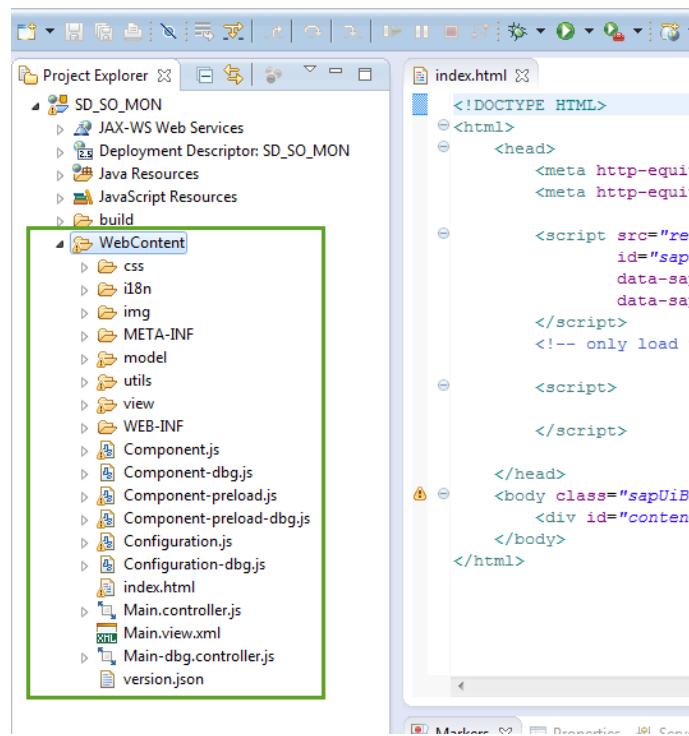
3.4.4. Select *SAPUI5 Application Development* → *Application project*, and click *Next*.



<p>3.4.5. Enter the project name: SD_SO_MON</p> <p>Uncheck the <i>Create an Initial View</i> check box.</p> <p>Click <i>Finish</i>.</p> <p><b>Note:</b> Keep the name of the project as SD_SO_MON here. Do not add any number to it.</p>	
<p>3.4.6. Right-click the <i>WebContent</i> folder and select <i>Import</i>.</p>	
<p>3.4.7. Select <i>General</i> → <i>File System</i> and select <i>Next</i>.</p>	

3.4.8. Click <i>Browse</i> .	
3.4.9. Select the <i>SD_SO_MON</i> folder from the desktop, and click <i>OK</i> .	
3.4.10. Select the project folder <i>SD_SO_MON</i> checkbox.  Click <i>Finish</i> .	

3.4.11. The standard application will be imported into Eclipse, as shown in the screenshot.

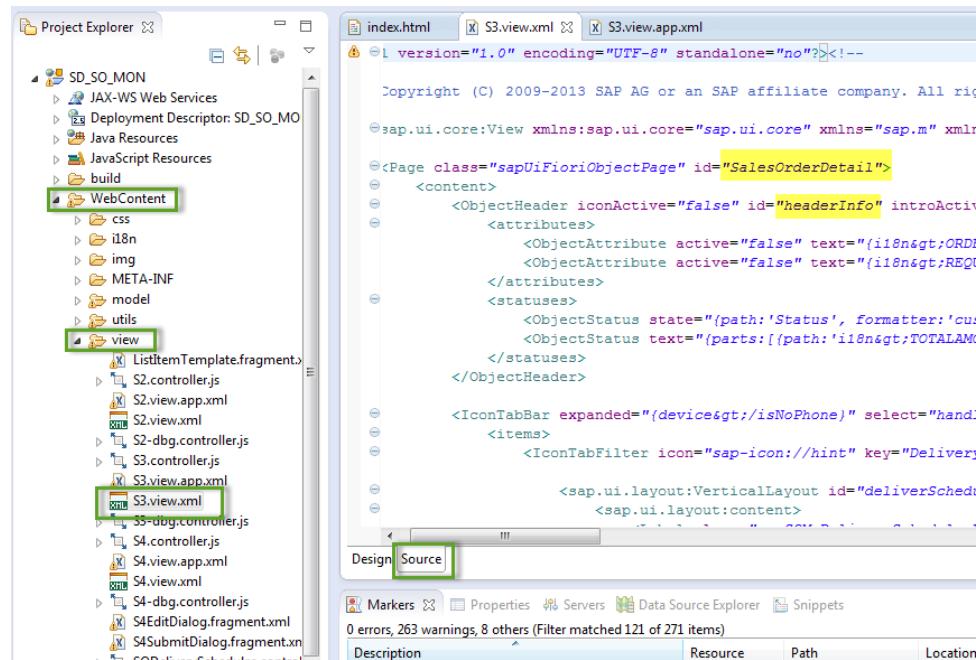


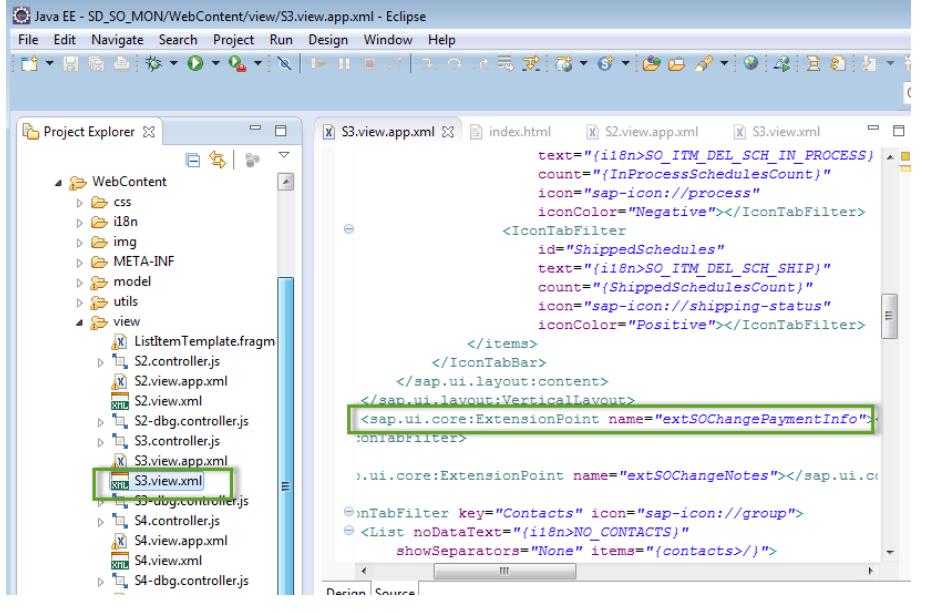
3.4.12. Each view inside the view folder is responsible for displaying some information.

First we need to find the view which displays the *Sales Order* header content.

The easiest way to find the corresponding view is to look for the relevant IDs related to Sales Order headers in the source code of each view.

*S3.view* has the source code to display the *Sales Order Details* view.



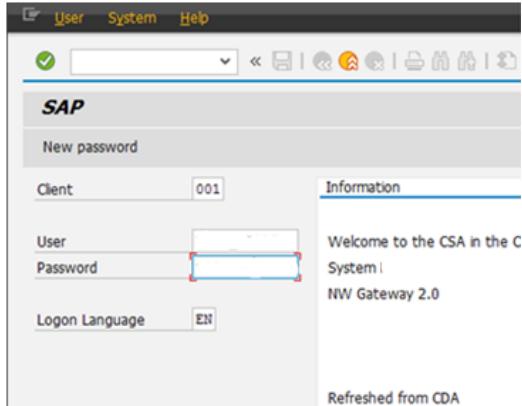
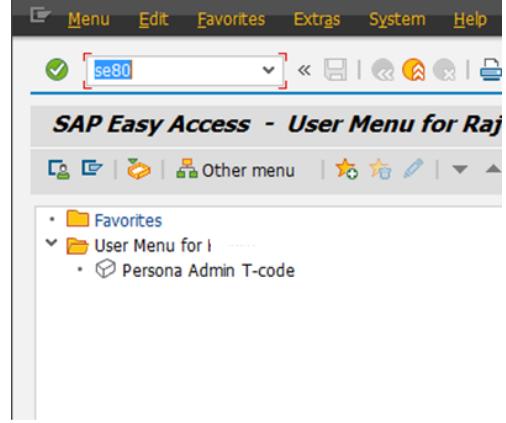
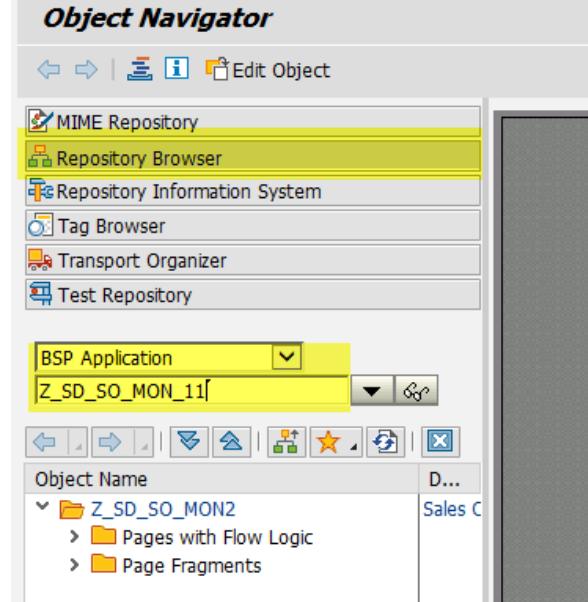
<p><b>3.4.13.</b> Now we need to find the extension point, which is used to display the custom information.</p> <p>All extension points are defined using the <code>&lt;core:ExtensionPoint&gt;</code> tag. The only way to find the extension point is to search for the <code>&lt;core:ExtensionPoint&gt;</code> tag in the source code of each view.</p>	<p><b>Sample Extension point declaration</b></p> <pre> &lt;Label text="{i18n&gt;REJECTION_STATUS}" /&gt; &lt;Text text="{path:'quotation&gt;/RejectionStatus', formatter:'cus.sd.myquotatio" /&gt;  &lt;Label text="{i18n&gt;REFSTATUS}" /&gt;&amp;nbsp;   &lt;Text text="{path:'quotation&gt;/ReferenceStatus' , formatter:'cus.sd.myquotat" /&gt;    &lt;core:ExtensionPoint name="extQuotationDetailsInfoBottom"/&gt; &lt;/form:content&gt; &lt;/form:SimpleForm&gt; &lt;/content&gt;&amp;nbsp; /IconTabFilter&gt;&amp;nbsp; </pre> <p><b>Note:</b> This screenshot is taken from different application for the explanation</p>
<p><b>3.4.14.</b> In the S3.view of our application we have an extension point named <code>extSOChangePaymentInfo</code>.</p> <p>We are going to use this extension point to display the <i>paymentInfo</i> (or any other custom information).</p> <p><b>Note:</b> if there is no extension point for a particular view or section then we have to use the property <code>sap.ui.viewReplacements</code> to replace the complete view (as we explained in point 1.4).</p>	 <p>The screenshot shows the Eclipse IDE interface with the Java EE - SD_SO_MON project open. The Project Explorer view on the left shows the structure of the application, including WebContent, css, i18n, img, META-INF, model, utils, and view folders containing various files like S3.view.xml, S3.controller.js, and S3.view.app.xml. The right side shows the S3.view.app.xml file in the editor. A specific line of code, <code>&lt;core:ExtensionPoint name="extSOChangePaymentInfo"&gt;</code>, is highlighted with a green rectangular selection. The code also includes other SAP Fiori XML elements like IconTabFilter, sap.ui.layout:content, and sap.ui.core:ExtensionPoint.</p>

<p>3.4.15. We also need to find out the section that displays material number, item, quantities, and price.</p>	
<p>3.4.16. In the <i>S4.view</i>, you can see the item detail information.</p> <p>The section id named <i>SO_OH_ITEMDETAIL</i> will display the item quantity and unit.</p> <p>Note the ID of the section. It will be used later to hide this section.</p>	

### 3.5. Deciding the Application Name

From now we are going to create new objects. First, think of two digit numbers as - mentioned in point 2.3.2 - which you are going to append to your objects.

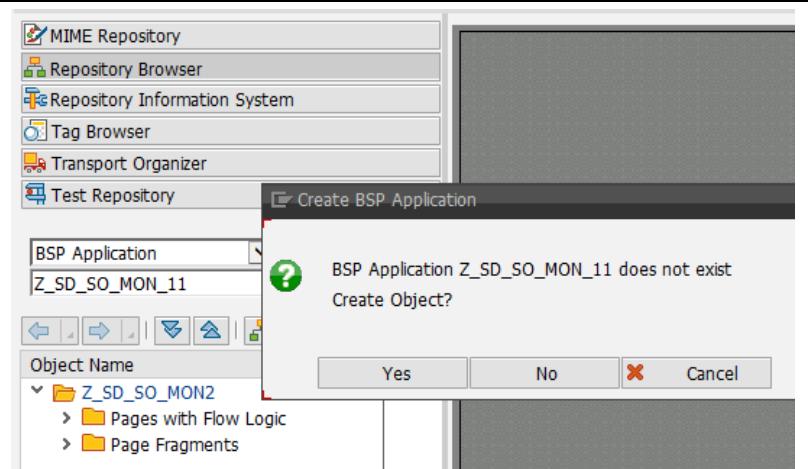
What to do	What to Say
------------	-------------

<p>3.5.1. Let's say you are going to append “_11” to all object names. First make sure that no one has already created a BSP application with the same name.</p> <p>The standard application name is SD_SO_MON. Therefore, you are going to create a new project as Z_SD_SO_MON_11. Make sure that there no other BSP projects called Z_SD_SO_MON_11.</p> <p>To begin doing this, login to the backend system.</p>	
<p>3.5.2. Run transaction SE80.</p>	
<p>3.5.3. Select the <i>Repository Browser</i>. Select the dropdown value, <i>BSP Application</i>.</p> <p>Enter the application name as Z_SD_SO_MON_11 (the name of the BSP application you are going to create).</p> <p>Press Enter.</p>	

If you get a message saying that BSP application does not exist, then you can proceed. However, now press cancel button.

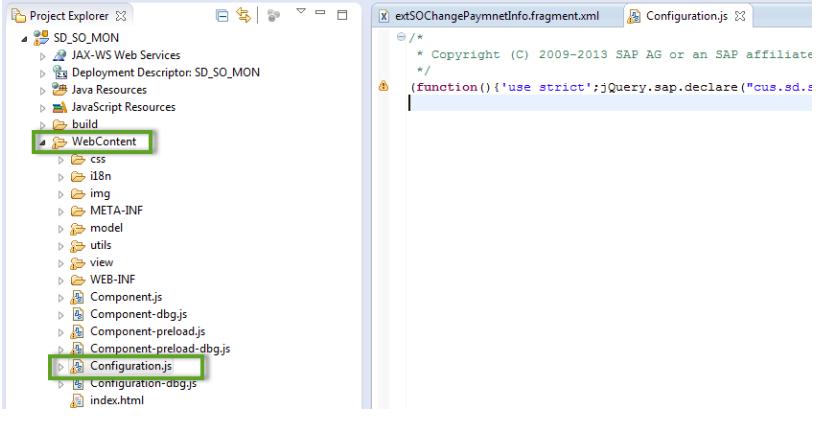
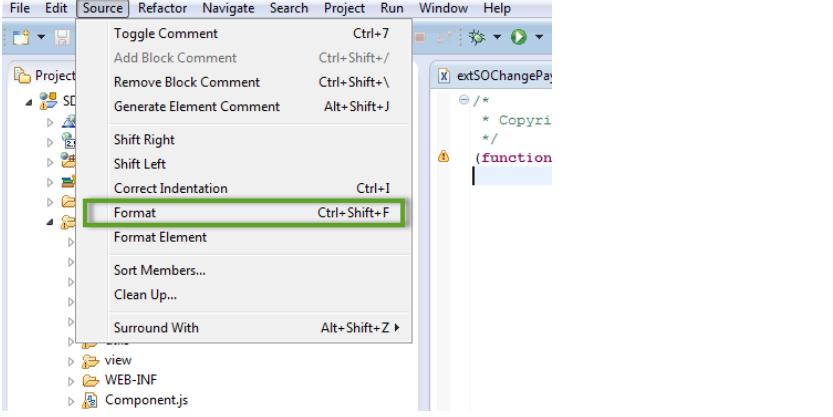
#### 3.5.4.

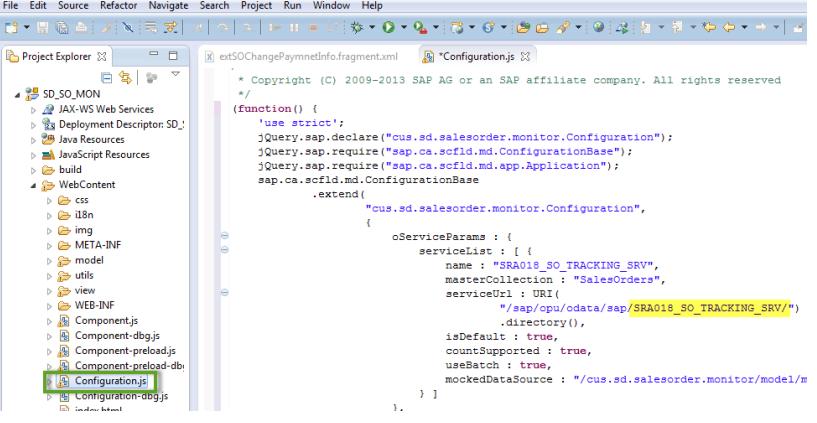
If there is a BSP application with the same name, change the name of your one to something like Z\_SD\_SO\_MON\_12, 13, 14... and so on.

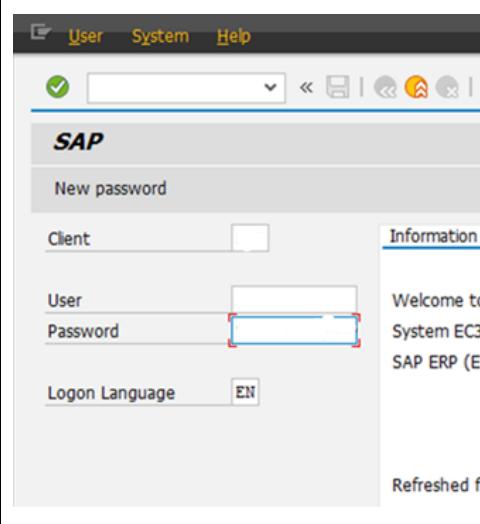
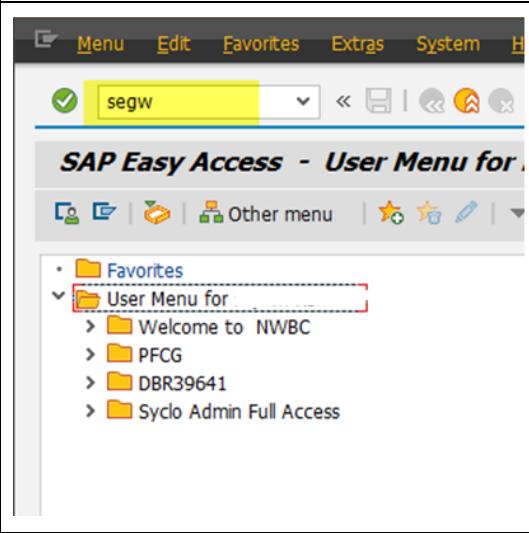
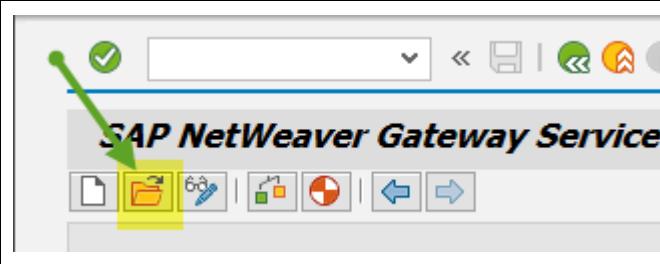
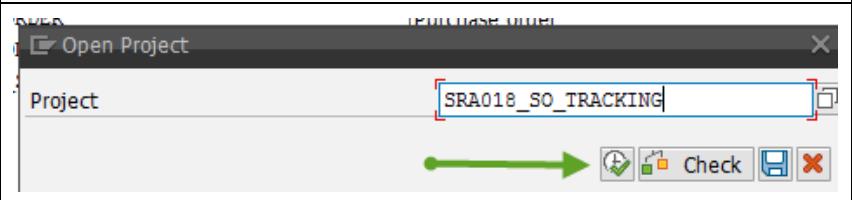


## 3.6. Gateway Extensibility

If we are using existing fields in the standard Gateway service, then we do not need to extend it. But if we need more information than the standard Gateway service provides, then we do need to extend it.

What to do	What to Say
3.6.1. In Eclipse, select the <i>Configuration.js</i> file in the <i>WebContent</i> folder.	
3.6.2. Select <i>Source</i> → <i>Format</i> .	

<p>3.6.3. In <i>Configuration.js</i>, we can see the standard Gateway service (highlighted).</p> <p>The gateway service used in this application is <i>SRA018_SO_TRACKING_SRV</i>.</p> <p>This gateway service features the <i>PaymentTerm</i> field. However, it does not have the logic to pull the data for the payment term field. We therefore need to extend the Gateway service to get the payment term information.</p>	 <pre> File Edit Source Refactor Navigate Search Project Run Window Help extSOChange@ymnetInfo.fragment.xml *Configuration.js Copyright (C) 2009-2013 SAP AG or an SAP affiliate company. All rights reserved /* (function() {     'use strict';     jQuery.sap.declare("cus.sd.salesorder.monitor.Configuration");     jQuery.sap.require("sap.ca.scfld.md.ConfigurationBase");     jQuery.sap.require("sap.ca.scfld.md.app.Application");     sap.ca.scfld.md.ConfigurationBase         .extend(             "cus.sd.salesorder.monitor.Configuration",             {                 oServiceParams : {                     serviceList : [                         {                             name : "SRA018_SO_TRACKING_SRV",                             masterCollection : "SalesOrders",                             serviceUrl : URL(                                 "/sap/cp/odata/sap/ZSRA018_SO_TRACKING_SRV/")                                 .directory(),                             isDefault : true,                             countSupported : true,                             useBatch : true,                             mockedDataSource : "/cus.sd.salesorder.monitor/model/n                 }             }         ]     ); }) </pre>
<p>3.6.4. For this scenario, the Gateway service, <i>SRA018_SO_TRACKING_SRV</i>, has been extended.</p> <p>The new Gateway service is <i>ZSRA018_SO_TRACKING_SRV</i>.</p> <p>Testing confirms that the new Gateway service is drawing information to the <i>PaymentTerms</i> field.</p> <p>Note the field name, <i>PaymentTerms</i>.</p> <p>If you want to extend the Gateway service yourself, please follow the next steps. Otherwise, go to section 3.7 to start building the Fiori application using the existing enhanced Gateway service.</p> <p>Replace &lt;your_ABAP_URL&gt; with <a href="http://&lt;ABAP server&gt;:&lt;port&gt;">http://&lt;ABAP server&gt;:&lt;port&gt;</a></p>	<p><b>Extended Gateway service complete URL</b>  <a href="http://&lt;your_ABAP_URL&gt;/sap/opu/odata/sap/ZSRA018_SO_TRACKING_SRV">http://&lt;your_ABAP_URL&gt;/sap/opu/odata/sap/ZSRA018_SO_TRACKING_SRV</a></p> <p><b>output of the enhanced gateway service</b></p> <pre> &lt;content type="application/xml"&gt; &lt;m:properties&gt;     &lt;d:SalesOrderNumber&gt;15962&lt;/d:SalesOrderNumber&gt;     &lt;d:Division /&gt;     &lt;d:DistributionChannel /&gt;     &lt;d:ShipToPostalCode /&gt;     &lt;d:ShipToRegion /&gt;     &lt;d:ShipToCountry /&gt;     &lt;d:PO&gt;89&lt;/d:PO&gt;     &lt;d&gt;TotalAmount&gt;120.00&lt;/d&gt;TotalAmount&gt;     &lt;d:Currency&gt;EUR&lt;/d:Currency&gt;     &lt;d&gt;Status&gt;&lt;/d&gt;Status&gt;     &lt;d:ShipToPartyName&gt;Becker Berlin&lt;/d:ShipToPartyName&gt;     &lt;d:OrderDate&gt;20140420&lt;/d:OrderDate&gt;     &lt;d:RequestedDate&gt;20140421&lt;/d:RequestedDate&gt;     &lt;d:ShipToContact /&gt;     &lt;d:ShipToAddressLine1&gt;Calvinstrasse&lt;/d:ShipToAddressLine1&gt;     &lt;d:ShipToAddressLine2&gt;Berlin 11 19467&lt;/d:ShipToAddressLine2&gt;     &lt;d:ShipToAddressLine3&gt;DE&lt;/d:ShipToAddressLine3&gt;     &lt;d&gt;TotalSchedulesCount&gt;2&lt;/d&gt;TotalSchedulesCount&gt;     &lt;d:OpenSchedulesCount&gt;2&lt;/d:OpenSchedulesCount&gt;     &lt;d&gt;InProcessSchedulesCount&gt;0&lt;/d&gt;InProcessSchedulesCount&gt;     &lt;d:ShippedSchedulesCount&gt;0&lt;/d:ShippedSchedulesCount&gt;     &lt;d:OrderItemsCount&gt;2&lt;/d:OrderItemsCount&gt;     &lt;d:PaymentTerms&gt;14 Days 3%, 30/24, 45 net&lt;/d:PaymentTerms&gt;     &lt;d:NetPriceAmount&gt;7120.00&lt;/d:NetPriceAmount&gt;     &lt;d:TaxAmount&gt;0.00000000&lt;/d:TaxAmount&gt;     &lt;d:ShipToCity /&gt;     &lt;d:CustomerID /&gt;     &lt;d:SalesOrganization /&gt;     &lt;d:CustomerName /&gt; &lt;/m:properties&gt; &lt;/content&gt; </pre>

<p>3.6.5. To begin extending the Gateway service, logon on to the backend system.</p> <p><b>Note:</b> To extend the Gateway service you need to have development access with your own User in both the backend system and the Gateway system.</p> <p>Please create an IT-Direct Ticket (if necessary) to get the development access in both system.</p>	
<p>3.6.6. Run transaction SEGW.</p>	
<p>3.6.7. Select Open.</p>	
<p>3.6.8. Enter SRA018_SO_TRACKING (SAP standard Gateway service name used in the Fiori application) and select Execute.</p>	

3.6.9. The Gateway service opens.

Select the *SalesOrder* entity type. Note the *PaymentTerms* field.

Even though there is a field for payment terms there is no logic to fetch the value for the field.

**SAP NetWeaver Gateway Service Builder**

The screenshot shows the SAP NetWeaver Gateway Service Builder interface. On the left, a tree view displays the project structure under 'SRA018\_SO\_TRACKING'. The 'SalesOrder' entity type is selected, highlighted with a yellow border. On the right, a table lists the properties of the SalesOrder entity, with 'PaymentTerms' also highlighted in yellow.

Properties	
•	SalesOrderNumber
•	ShipToPostalCode
•	ShipToRegion
•	ShipToCountry
•	PO
•	TotalAmount
•	Currency
•	Status
•	ShipToPartyName
•	OrderDate
•	RequestedDate
•	ShipToContact
•	ShipToAddressLine1
•	ShipToAddressLine2
•	ShipToAddressLine3
•	TotalSchedulesCount
•	OpenSchedulesCount
•	InProcessSchedulesCount
•	ShippedSchedulesCount
•	OrderItemsCount
•	PaymentTerms
•	NetDueAmount

3.6.10. Select the *Create* icon to begin creating the new extended project.

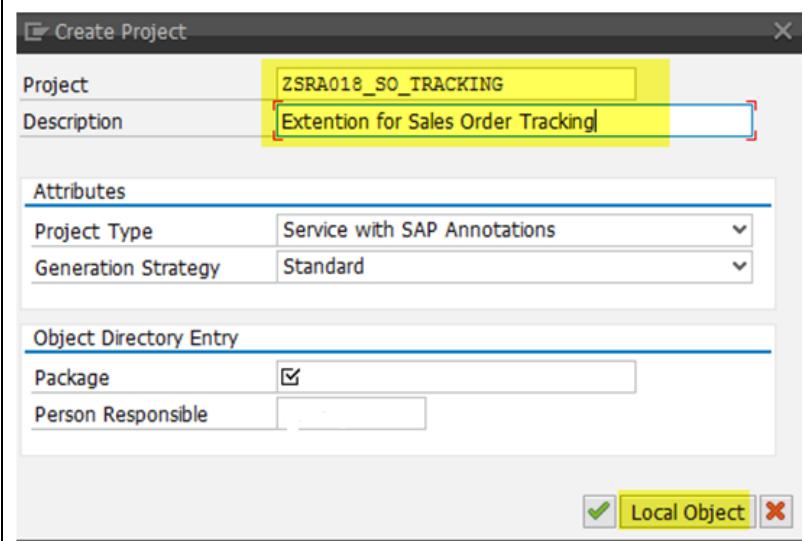
**SAP NetWeaver Gateway Service Builder**

The screenshot shows the SAP NetWeaver Gateway Service Builder interface with a list of services on the right. A green arrow points to the 'Create' icon in the toolbar at the top.

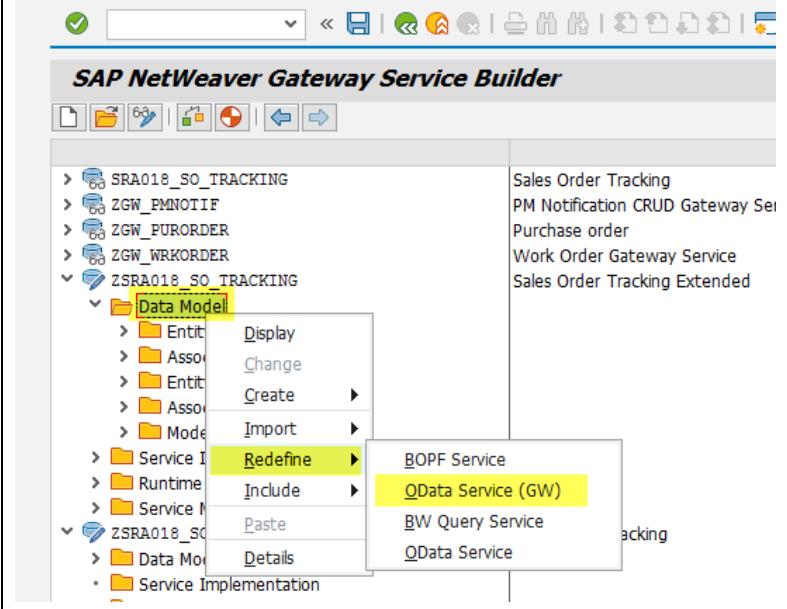
Services	
>	SRA018_SO_TRACKING
>	ZGW_PMNOTIF
>	ZGW_PURORDER
>	ZGW_WRKORDER

- 3.6.11. Enter the new project name. Click *Local Object* if you don't wish to transport this change. Otherwise assign it to an existing package.

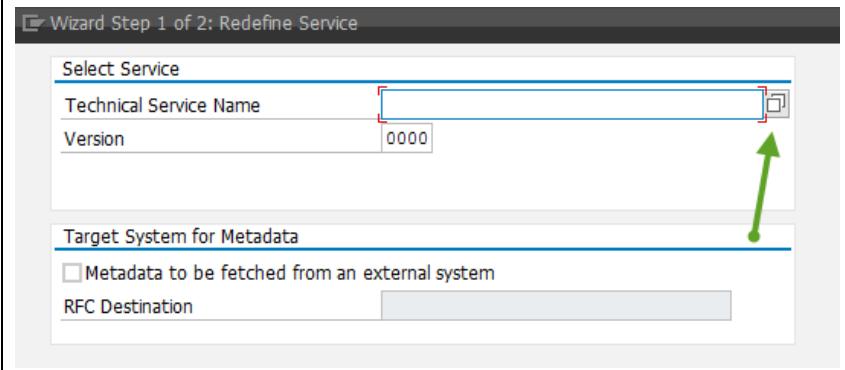
**Note:** Keep your naming convention.  
ZRSA018\_SO\_TRACKING\_XX (where XX is the number).



- 3.6.12. Right-click *Data Model* and select *Redefine* → *OData Service (GW)*.



- 3.6.13. On the first page of the wizard, select the F4 help as indicated.



3.6.14. Select the *SRA018\_SO\_TRACKIGN\_SRV* and select the green tick mark icon.

Technical Service Name	Version	Technical Model Name	Mdl. Vers.
SRA007_BENEFITS_SRV	1	SRA007_BENEFITS_MDL	1
SRA008_SRV	1	SRA008_MDL	1
SRA009_SRV	1	SRA009_MDL	1
SRA010_TIMESHEET_APPROVAL_SRV	1	SRA010_TIMESHEET_APPROVAL_MDL	1
SRA012_SPEND_SRV	1	SRA012_SPEND_MDL	1
SRA013_PO_FROM_PR_SRV	1	SRA013_PO_FROM_PR_MDL	1
SRA016_PRICE_AVAIL_SRV	1	SRA016_PRICE_AVAIL_MDL	1
SRA017_SALESORDER_CREATE_SRV	1	SRA017_SALESORDER_CREATE_MDL	1
<b>SRA018_SO_TRACKING_SRV</b>	<b>1</b>	<b>SRA018_SO_TRACKING_MDL</b>	<b>1</b>
SRA019_SHIPMENT_TRACKING_SRV	1	SRA019_SHIPMENT_TRACKING_MDL	1
SRA020_PO_TRACKING_SRV	1	SRA020_PO_TRACKING_MDL	1
SRA021_SRV	1	SRA021_MDL	1
SRA_SO_TRACKING_SRV	1	SRA_SO_TRACKING_MDL	1
SRM_PURCHASEORDER	1	SRM_PURCHASEORDER	1
S_EPM_SADL_GW_DEV_SCEN_RO_SRV	1	S_EPM_SADL_GW_DEV_SCEN_RO_MDL	1
VBI_APPL_DEF_SRV	1	VBI_APPL_DEF_MDL	1
VBI_GEOCODER_SRV	1	VBI_GEOCODER_MDL	1
WFA_WORKFLOW_SHIFT_PLAN	1	WFA_WORKFLOW_SHIFT_PLAN	1
Z1SRA020_PO_TRACKING_SRV	1	Z1SRA020_PO_TRACKING_MDL	1

3.6.15. Select *Next*.

Select Service

Technical Service Name: SRA018\_SO\_TRACKING\_SRV  
Version: 0001

Target System for Metadata

Metadata to be fetched from an external system

RFC Destination:

3.6.16. Click the *Select All* icon, and select *Finish*.

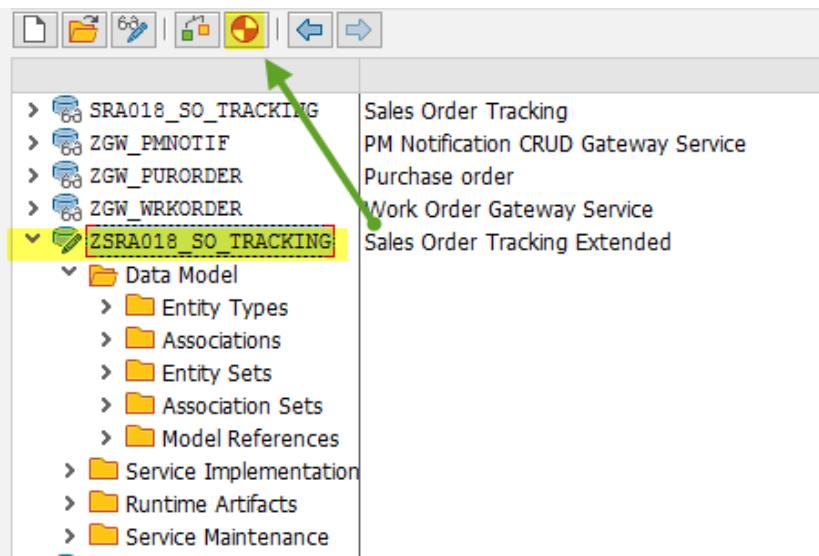
Service

OData Artifacts | Description

SRA018\_SO\_TRACKING\_MDL

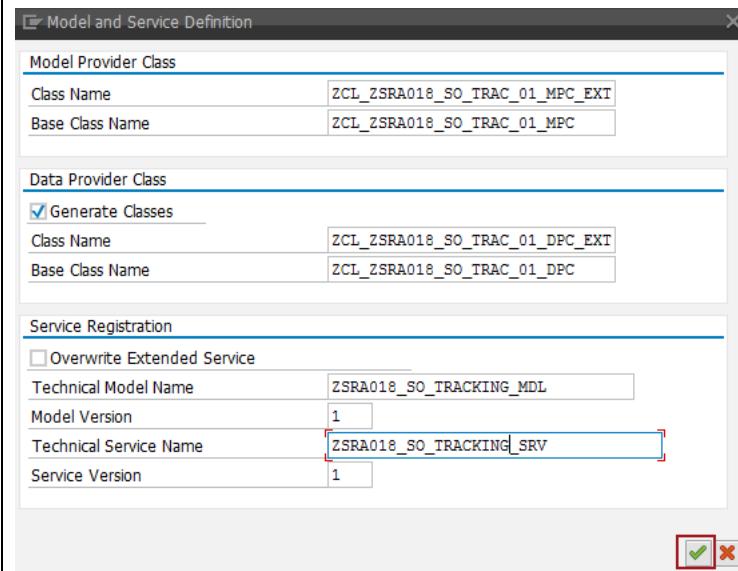
- >  SalesOrder Entity Type SalesOrder
- >  OrderItem Entity Type OrderItem
- >  ScheduledItem Entity Type ScheduledItem
- >  Attachment Entity Type Attachment
- >  Contact Entity Type Contact
- >  Country Entity Type Country
- >  Customer Entity Type Customer
- >  SalesOrderOrderItem Association SalesOrderOrderItem
- >  SalesOrderAttachment Association SalesOrderAttachment
- >  SalesOrderScheduledItem Association SalesOrderScheduledItem
- >  OrderItemScheduledItem Association OrderItemScheduledItem

- 3.6.17. Select the new Gateway service, and click the *Generate* icon.

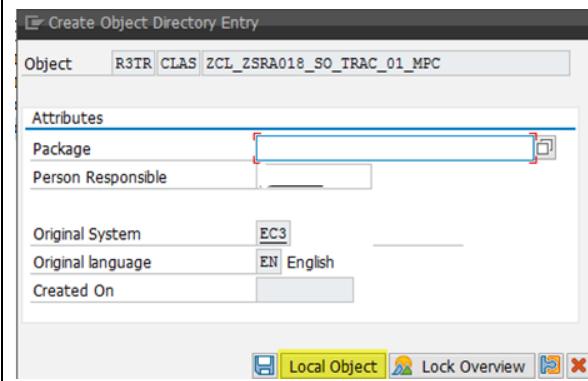


- 3.6.18. Select the green tick mark in the *Model and Service Definition* window.

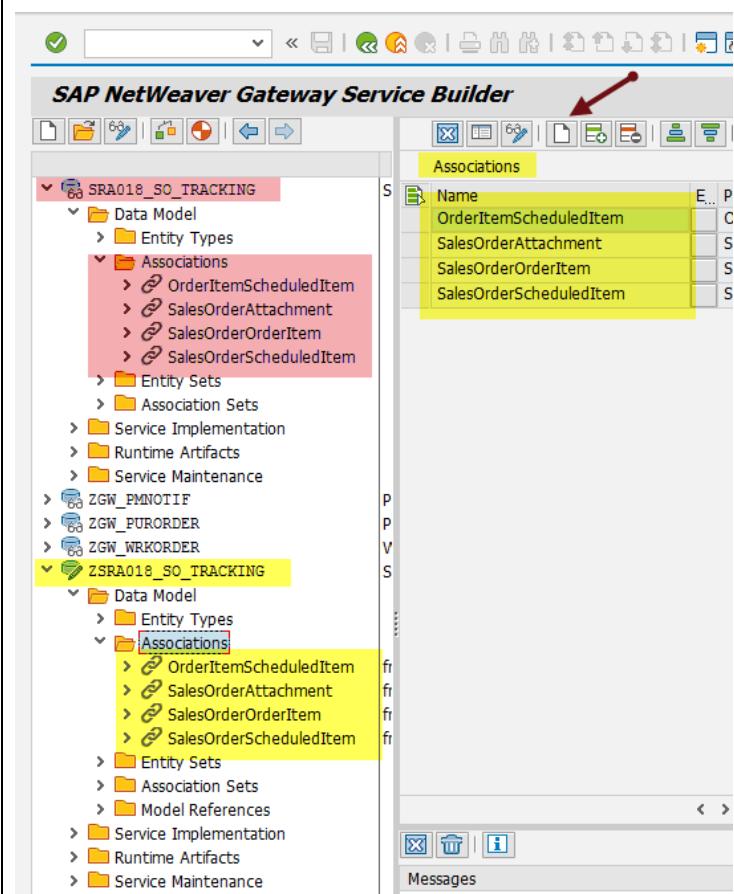
**Note:** Do not change any of the object names here.



- 3.6.19. Click *Local Object* if you don't wish to transport this change, otherwise assign it to an existing package.

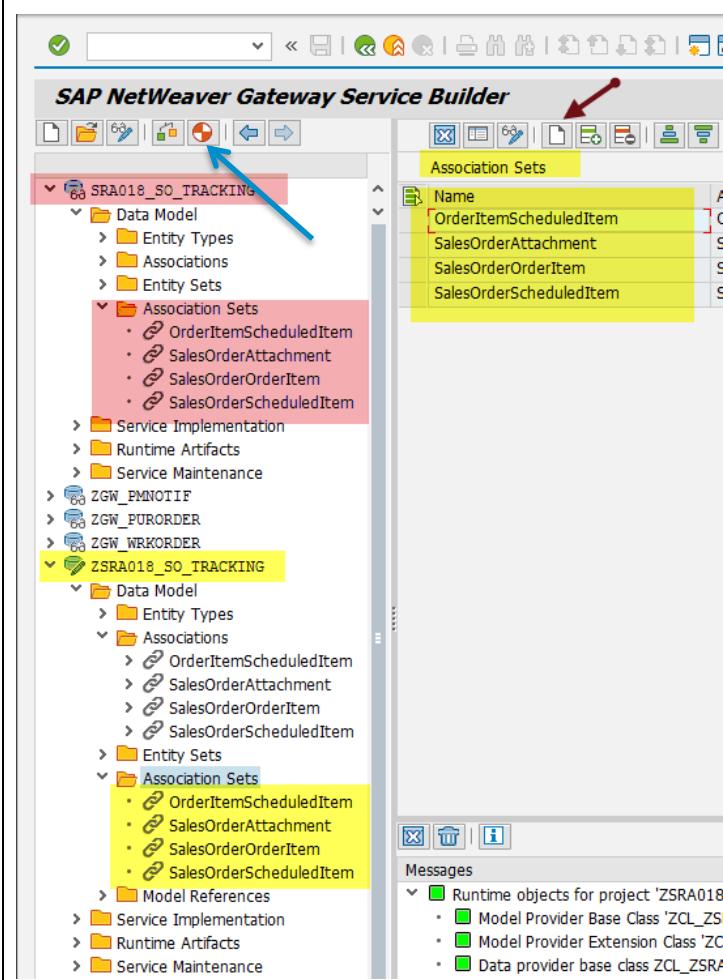


3.6.20. Create all the associations manually in the new Gateway service (highlighted in yellow) by using the *Add Association* icon in the *Associations* screen. Create the same associations as in the standard Gateway service (highlighted in red).

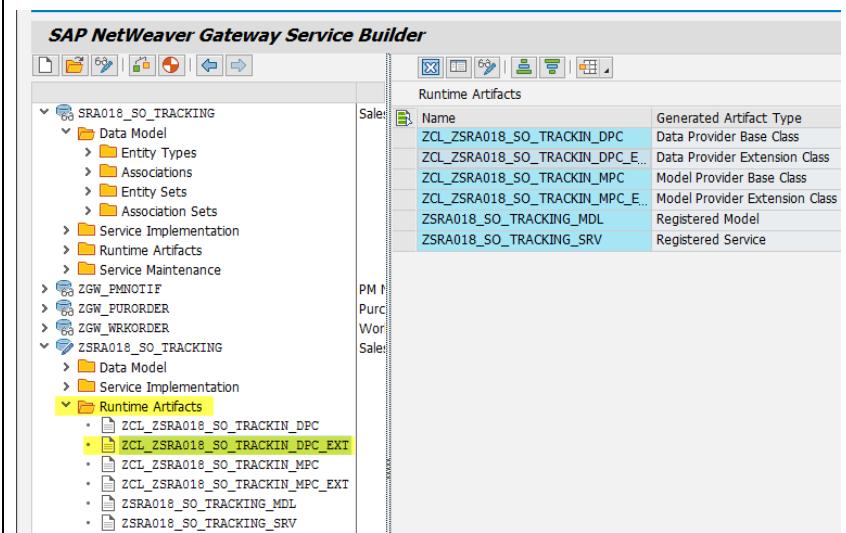


3.6.21. Create all the association sets manually in the new Gateway service (highlighted in yellow) by using the *Add Association Set* icon in the *Association Sets* screen. Create the same association sets as in the standard Gateway service (highlighted in red).

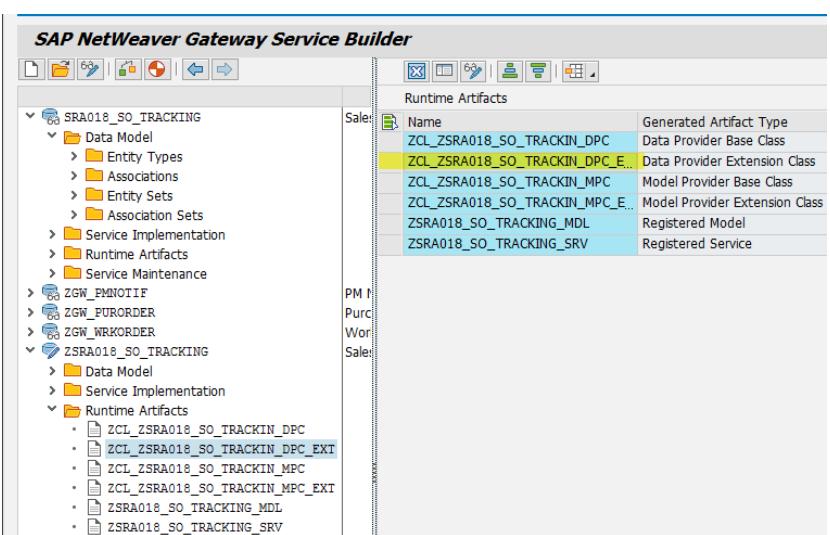
Note: Click the *Generate* icon once done.



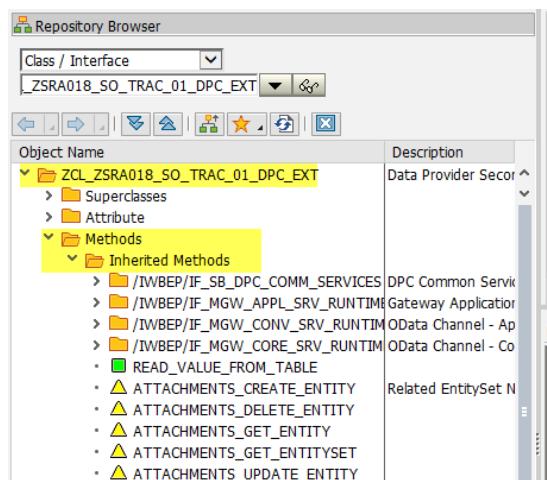
3.6.22. Double-click the highlighted *DPC\_EXT* class in the *Runtime Artifacts* folder.



3.6.23. Double-click the highlighted class.



3.6.24. If the left side navigator, expand the *Methods* and *Inherited Methods* nodes.

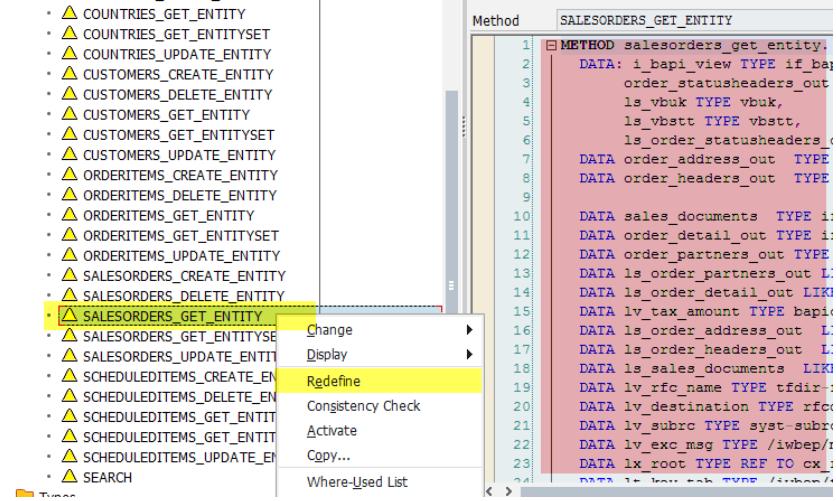


3.6.25. Find the method called *SALESORDERS\_GET\_ENTITY*. Right-click *SALESORDERS\_GET\_ENTITY* and then select *Redefine*.

**Note:** The section highlighted in red is responsible for retrieving details of the particular sales order.

Currently there is no business logic to return the payment terms.

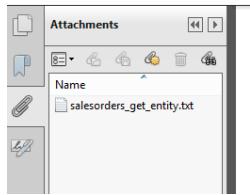
Therefore, we are going to redefine the method so it includes payment term information.



3.6.26. Once you select *Redefine*, the

*SALESORDERS\_GET\_ENTITY* method is displayed in the *Redefinitions* folder.

Remove all the lines in the highlighted section (highlighted in red) and copy the source code from the *salesorders\_getentity.txt* file attached to this pdf document to highlighted section.



**Note:** The new source code contains the logic to fetch the payment term information.

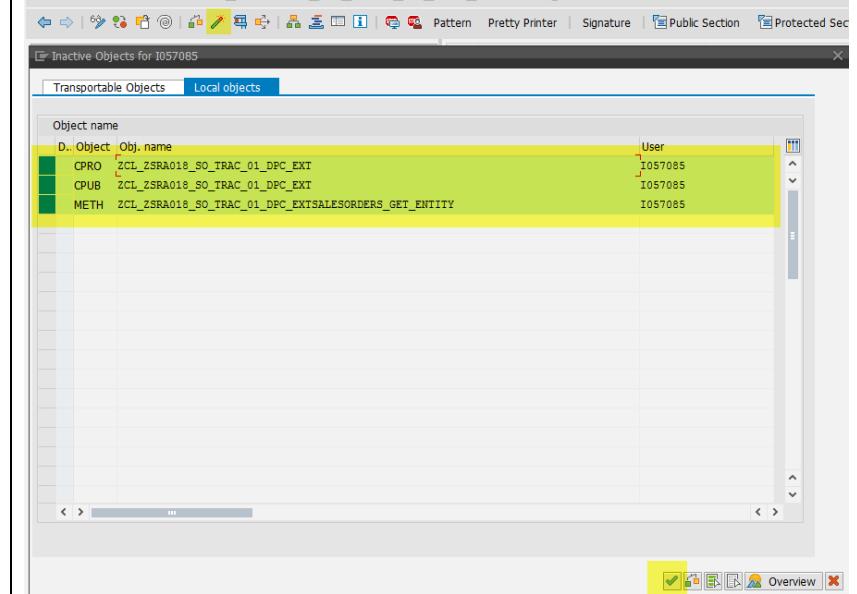
3.6.27. Click the *Activate* button and select all the objects.

Then, click the green tick mark icon to activate all the objects.

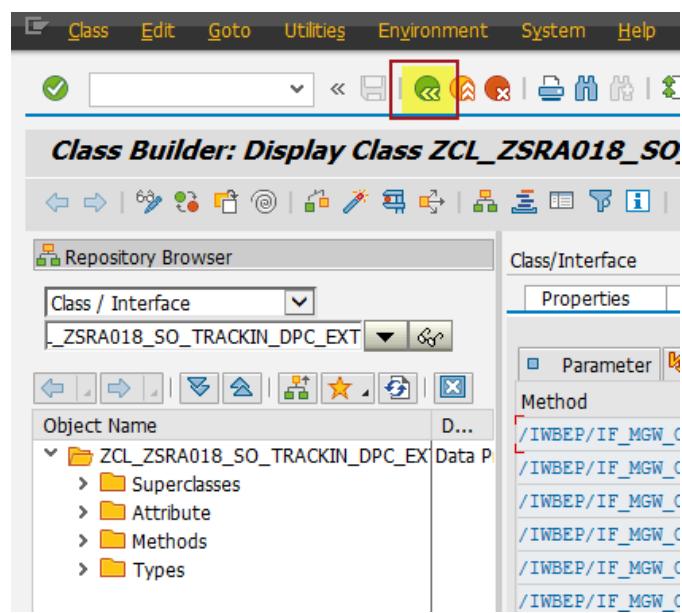
The screenshot shows the SAP GUI interface with two main panes. The left pane displays a tree view of methods under a node labeled 'Redefinitions'. The 'SALESORDERS\_GET\_ENTITY' method is highlighted in red. The right pane shows a detailed view of the 'SALESORDERS\_GET\_ENTITY' method in the 'Method Editor'. The code is as follows:

```
Method SALESORDERS_GET_ENTITY
method SALESORDERS_GET_ENTITY.
  *TRY.
  CALL METHOD SUPER->SALESORDERS_GET_ENTITY
  EXPORTING
    IV_ENTITY_NAME      =
    IV_ENTITY_SET_NAME  =
    IV_SOURCE_NAME      =
    IT_KEY_TAB          =
  ** io_request_object      =
  ** io_tech_request_context =
  ** IT_NAVIGATION_PATH    =
  IMPORTING
    er_entity            =
  .
  ** CATCH /iwbep/cx_mgv_busi_exception .
  ** CATCH /iwbep/cx_mgv_tech_exception .
ENDTRY.
endmethod.
```

*Class Builder: Class ZCL\_ZSRA018\_SO\_TRAC\_01\_DPC\_EXT Change*

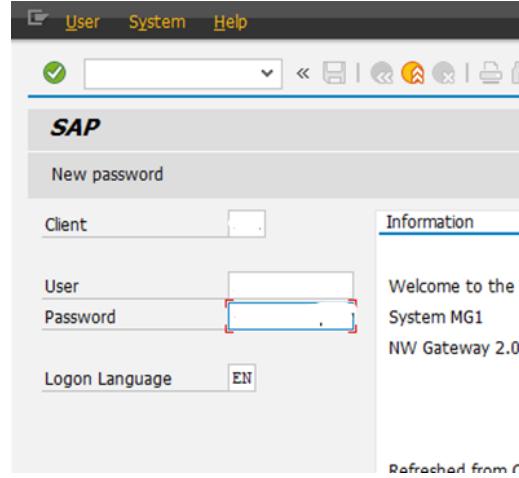


3.6.28. Click the *Back* icon to return to transaction SEGW.

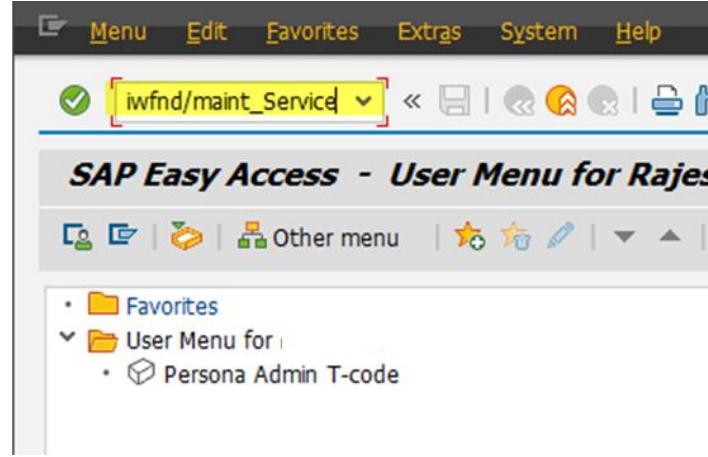


3.6.29. Now the new Gateway service has to be registered on the Gateway system.

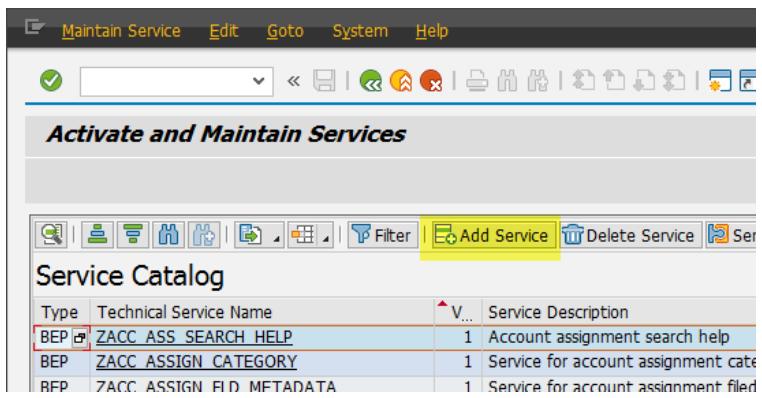
To do this, logon to the Gateway system.



3.6.30. Run transaction /IWFND/MAINT\_SERVICE.

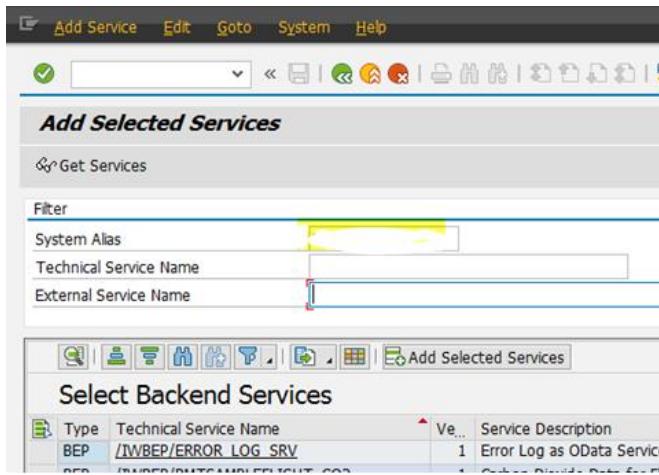


3.6.31. Click Add Service.



3.6.32. Enter the system Alias as xxx\_xxx and press *Enter*.

You can see the list of services from xxx System.



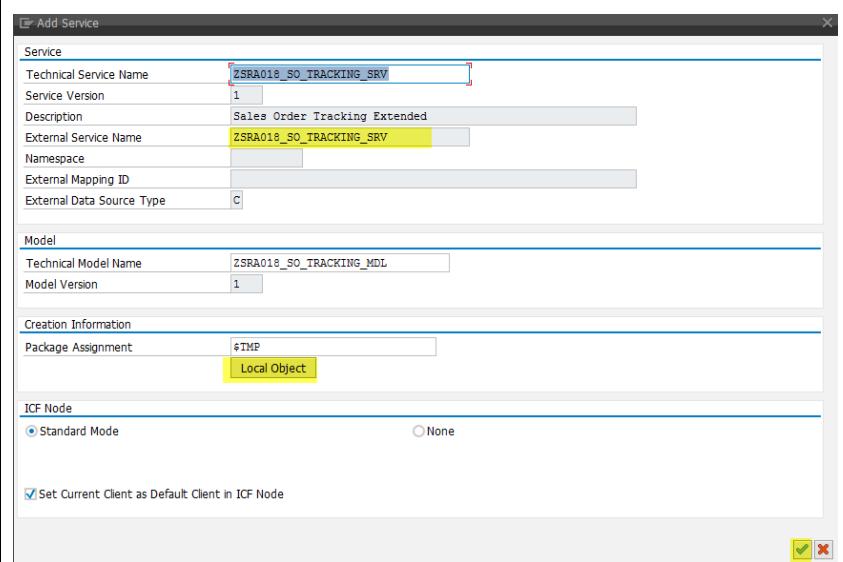
3.6.33. Find your Gateway service.

Double-click the Gateway service on the highlighted line. (In your case, double-click your Gateway service).

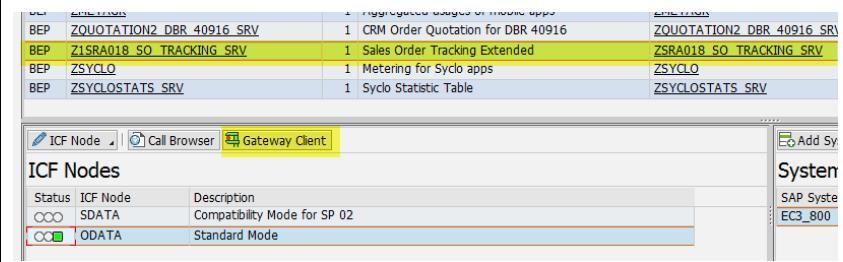
BEP	EPM_OIA_GW_SALES_HISTORY_SRV	1	CL_EPM_OIA_GW_SALES_HI_DPC_EXT	EPM_OIA_GW_SALES_HISTORY_SRV
BEP	EPM_OIA_SMOKETEST_SRV	1	OData Service for EPM OIA Smoketest	EPM_OIA_SMOKETEST_SRV
BEP	IRWC	1	ILM Reporting Work Center	IRWC
BEP	KP_LEAVE_REQUEST	1		KPLEAVEREQUEST
BEP	LWM_RS_REALSPEND	2	Real Spend Service	LWM_RS_REALSPEND
BEP	LWMSD_ORDERSTATUS_FIN	1	SAP ERP Order Status - Intergration with FIN	LWMSD_ORDERSTATUS_FIN
BEP	MDGEE	1	MDGee Service	MDGEE
BEP	MDGEE_DEV	1	MDGee Service (dev)	MDGEE_DEV
BEP	S_EPM_SADL_GW_DEV_SCEN_RO_SRV	1	EPM: SADL-based GW-Service 'Developer Scenario'	S_EPM_SADL_GW_DEV_SCEN_RO_SRV
BEP	SETTLEMENTREQUEST	1	Single Settlement Request	SETTLEMENTREQUEST
BEP	VBI_APPL_DEF_SRV	1	CL_VBL_APPL_DEF_01_DPC_EXT	VBI_APPL_DEF_SRV
BEP	VBI_GEOCODER_SRV	1	GeoCoding Service	VBI_GEOCODER_SRV
BEP	WFA_WORKFLOW_SHIFT_PLAN	1	Generic workflow service in WFM	WFA_WORKFLOW_SHIFT_PLAN
BEP	ZEPHSS_DOCUMENT_SERVICE	1	Data provider for PSS documents	ZEPHSS_DOCUMENT_SERVICE
BEP	ZSRA018_SO_TRACKING_SRV	1	Sales Order Tracking Extended	ZSRA018_SO_TRACKING_SRV
BEP	ZSRA018_SO_TRACKING2_SRV	1	Sales order Tracking	ZSRA018_SO_TRACKING2_SRV
BEP	ZSRA020_PO_TRACKING_SRV	1	PO Tracking Extensibility	ZSRA020_PO_TRACKING_SRV
BEP	ZUI5_THEMEDELETE_SRV	1	UI5 Custom Theme Deletion	ZUI5_THEMEDELETE_SRV

3.6.34. Click *Local Object* if you don't wish to transport this change. Otherwise assign it to an existing package and then select the green tick mark icon.

**Note:** The *External Service Name* will be the final Gateway service which will be exposed outside. (We cannot edit the external service name).



3.6.35. To test the modified service, select the service and then click the *Gateway Client* button.



3.6.36. In the *Request URI* field, append the text `SalesOrders('16000')` to the service.

Click the *Execute* button.

You can see OData getting executed and the output returning in xml format.

You can also see the payment term field being filled in xml format.

**Note:** Please enter the sales order number in the webservice that already exists in the system.

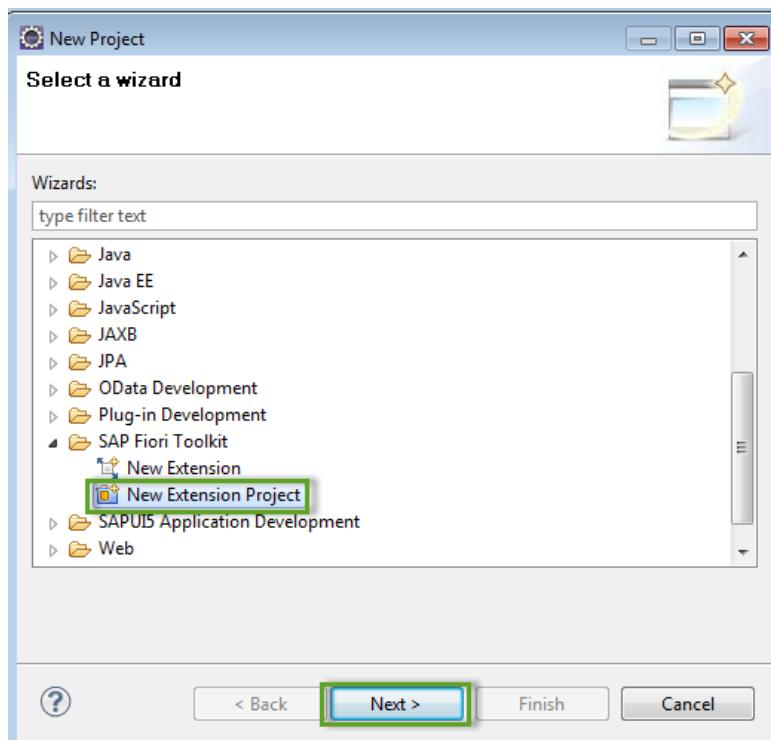
The screenshot shows the SAP NetWeaver Gateway Client interface. At the top, it displays the URL `/sap/opu/odata/sap/ZSRA018_SO_TRACKING_SRV/MAH_SRV/SalesOrders('16000')`. Below the URL, the 'HTTP Response' section shows the XML output of the OData query. The XML includes fields like `~status_code` (200), `~status_reason` (OK), `~server_protocol` (HTTP/1.0), and `set-cookie` (sap-usercontext=sap-client=004; path=/). The main content of the response is a sales order with details such as `TotalAmount` (8590.00), `Currency` (EUR), `Status` (A), `ShipToPartyName` (Becker Berlin (Versand)), `OrderDate` (20140518), `RequestedDate` (20140519), `ShipToContact`, `ShipToAddressLine1` (Beatestrasse), `ShipToAddressLine2` (Berlin 11 13505), `ShipToAddressLine3` (DE), `TotalSchedulesCount` (1), `OpenSchedulesCount` (1), `InProcessSchedulesCount` (0), `ShippedSchedulesCount` (0), `OrderItemsCount` (1), `PaymentTerms` (14 Days 3%, 30/2%, 45 net), `NetPriceAmount` (8590.00), `TaxAmount` (0.00000000), `ShipToCity`, `CustomerID`, `SalesOrganization`, `CustomerName`, and `m:properties`.

## 3.7. Creating a New Enhancement SAPUI5 Project

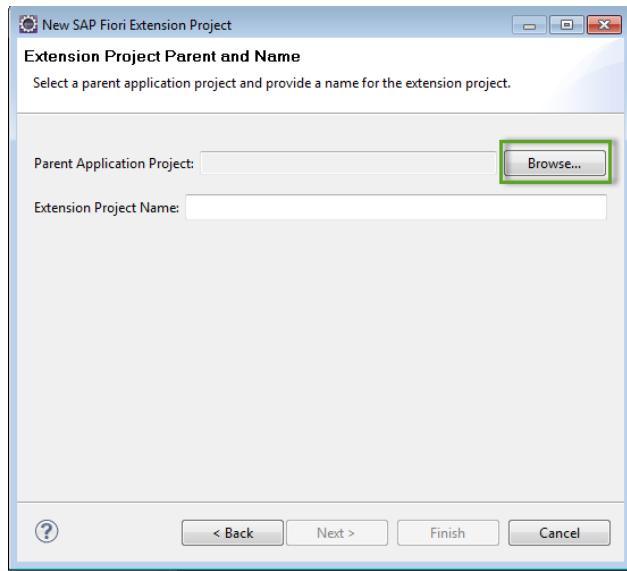
You need to create a new enhancement project using the *SAP Fiori Toolkit* where you can add or modify the existing views. By doing this, you can achieve the goal without changing the existing content.

What to do	What to Say
3.7.1. In Eclipse, select <i>File</i> → <i>New</i> → <i>Project</i> .	<p>The screenshot shows the Eclipse 'File' menu open. The 'New' option is highlighted. To the right of the menu, there is a list of project types: JPA Project, Enterprise Application Project, Dynamic Web Project, EJB Project, Connector Project, Application Client Project, Static Web Project, Project... (with an arrow pointing to it), Servlet, and Session Bean (EJB 3.x).</p>

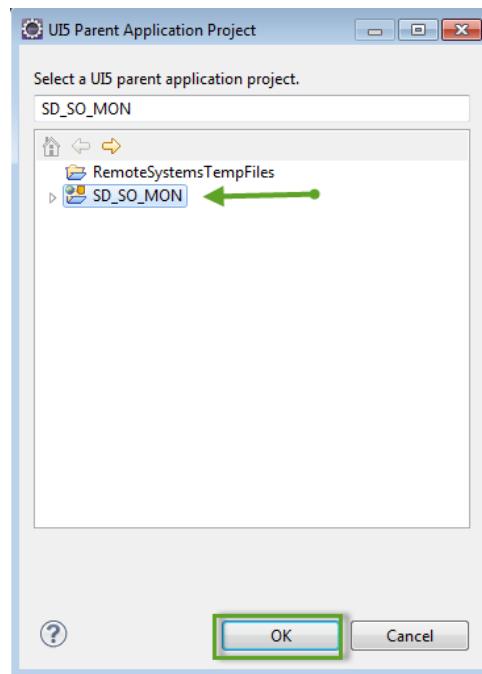
3.7.2. Select *New Extension Project* in the SAP Fiori Toolkit and choose *Next*.



3.7.3. Click *Browse* to select the standard application as parent project.



- 3.7.4. Select the project *SD\_SO\_MON* and choose *OK*.

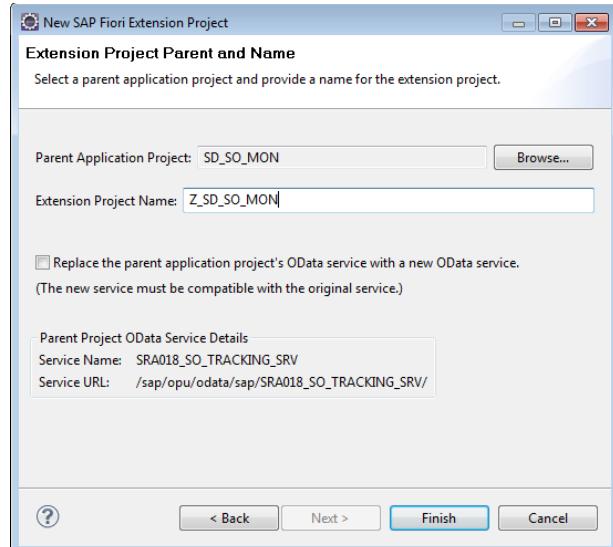


- 3.7.5. In the *Extension Project Name* field, enter *Z\_SD\_SO\_MON*.

Choose *Finish*.

**Note:** You can add the new extended Gateway service manually later.

Create the project and call it *Z\_SD\_SO\_MON\_XX* (where XX is the number). This is because the project *Z\_SD\_SO\_MON* already exists in the system.



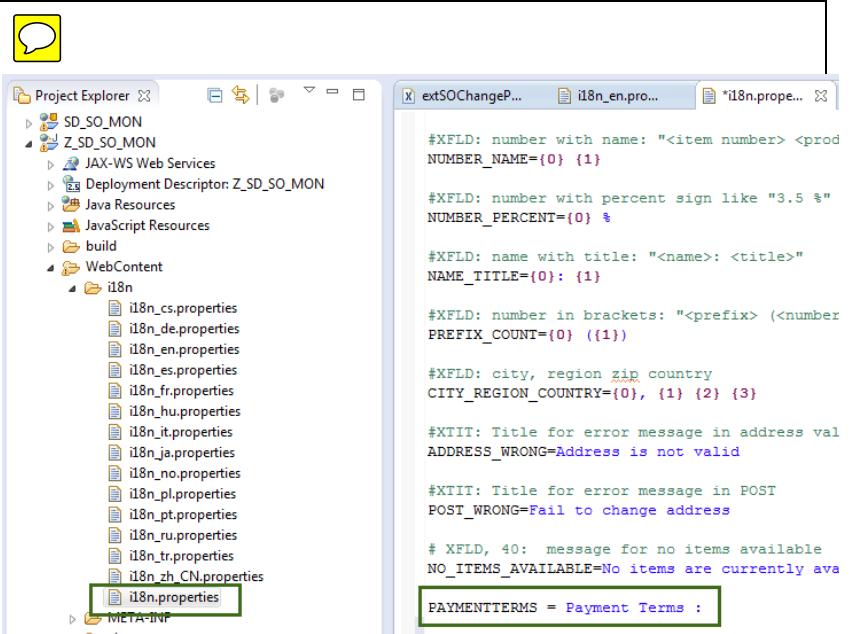
### 3.7.6. The *i18n* folder contains the translation files of the application.

Add the payment term description in the *i18n.properties* file.

If you are using different language, please add the respective description in the respective language file.

All the translations will be picked up from here

Note: Click on the  icon to copy the source code.



The screenshot shows the Eclipse IDE's Project Explorer and a code editor. The Project Explorer displays a hierarchy of projects and their contents, including the *i18n* folder which contains multiple language properties files like *i18n\_en.properties*, *i18n\_de.properties*, etc. The code editor shows the *i18n\_en.properties* file with some key-value pairs and the *i18n.properties* file which contains the new entry **PAYMENTTERMS = Payment Terms :**.

```
#XFLD: number with name: "<item number> <prod NUMBER_NAME={0} {1}

#XFLD: number with percent sign like "3.5 %"
NUMBER_PERCENT={0} %

#XFLD: name with title: "<name>: <title>"
NAME_TITLE={0}: {1}

#XFLD: number in brackets: "<prefix> (<number PREFIX_COUNT={0} {(1)} {2} {3}

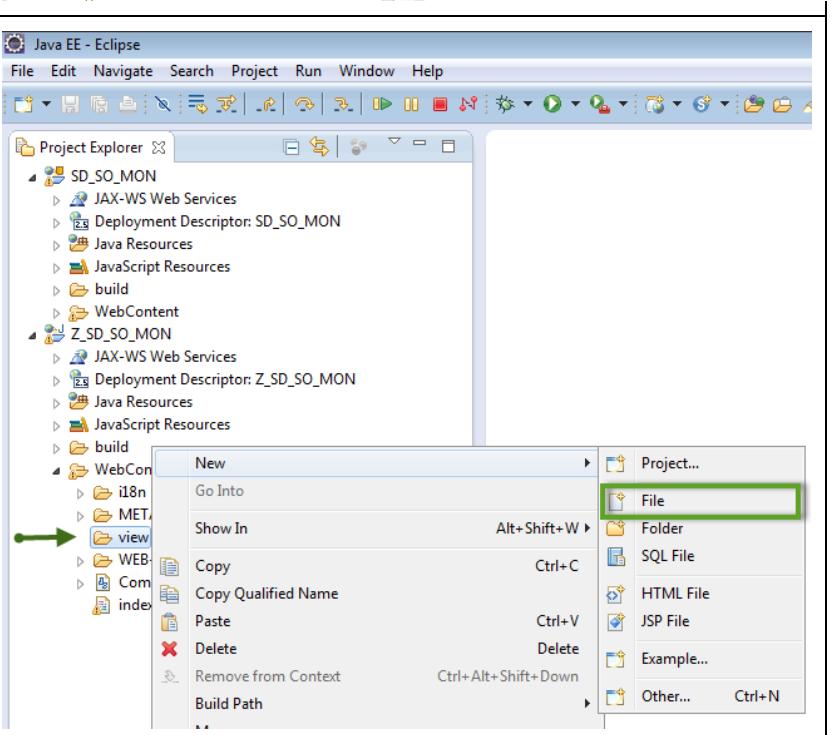
#XTIT: Title for error message in address val
ADDRESS_WRONG=Address is not valid

#XTIT: Title for error message in POST
POST_WRONG=Fail to change address

# XFLD, 40: message for no items available
NO_ITEMS_AVAILABLE=No items are currently ava

PAYMENTTERMS = Payment Terms :
```

### 3.7.7. In the extended project, right-click the *View* folder. Choose *New* → *File*.

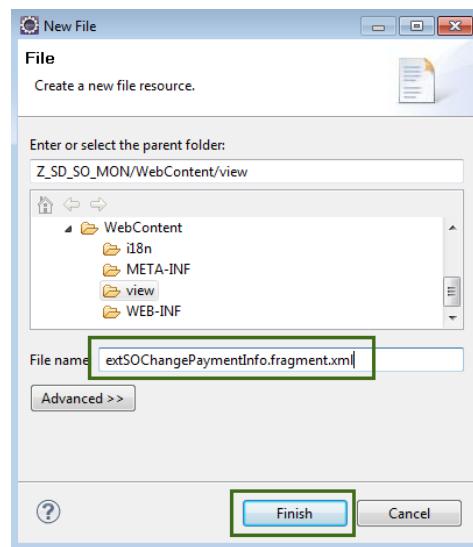


- 3.7.8. Enter the file name as `extSOChangePaymentInfo.fragment.xml`.

Click *Finish*.

You are going to add your custom fields in this fragment and then link it to the extension point later.

**Note:** Keep the same name as shown in the screenshot.



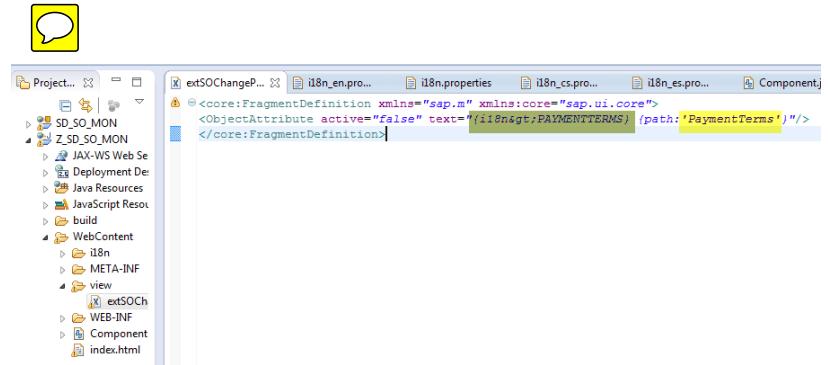
- 3.7.9. You have to add your custom fields inside the *FragmentDefinition* tag.

Add the payment term field inside the *FragmentDefinition* tag.

The text highlighted in yellow is the name of *Payment Term* field. This should be same as the field name you declared in the Gateway service (see Point 3.5.4).

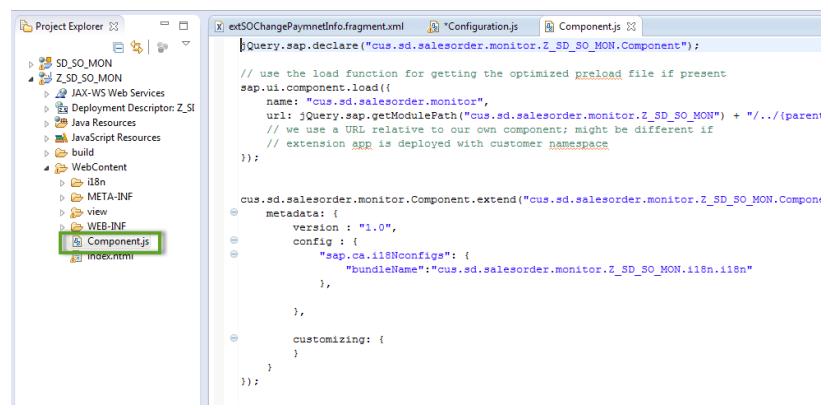
The text highlighted in green is the label for the *Payment Term* field we declared in the *i18n* file(see Point 3.6.6)

**Note:** Click the  icon to copy the source code.



- 3.7.10. Now you have to link the fragment to the extension point.

Select the *Component.js* file in the *WebContent* folder.



### 3.7.11. Modify the code in *Component.js* as shown in the screenshot.

Click *Save All*.



#### Source Code Explanation

##### Section Highlighted in Yellow

We are declaring a new module path. Each Fiori application has a specific module path. As it is a custom application, we are declaring with Z namespace.

##### Section highlighted in Green

This section is the reference to the standard application.

##### Section highlighted in Grey

Declaration for including the new changes.

##### Section highlighted in Blue

We are including our custom gateway service. This has to be included to create a new, enhanced gateway service.

This section also contains the new Gateway service name. If you created your new Gateway service please replace the text *ZSRA018\_SO\_TRACKING\_SRV* with your Gateway service name. (Refer to points 3.6.35 or 3.6.36 for the external service name).

##### Section Highlighted in Orange

In this section we are linking fragment to the EnhancementPoint. (Refer to point 3.4.15).

##### Section Highlighted in Red

We are also making the container *SO\_OH\_ITEMDETAIL* invisible in the S4view (Refer to point 3.4.17).

**Note:** Click the  icon to copy the source code.



```

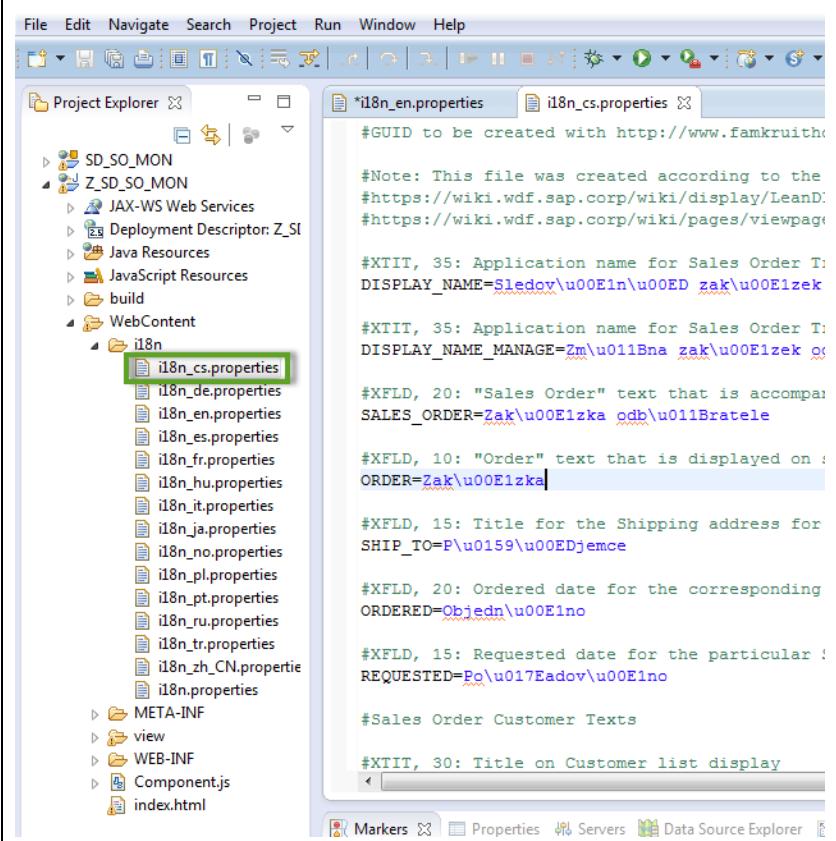
jQuery.sap.declare("zcus.sd.salesorder.monitor.Component");

// use the load function for getting the optimized preLoad file if present
sap.ui.component.load({
    name: "zcus.sd.salesorder.monitor",
    url: jQuery.sap.getModulePath("zcus.sd.salesorder.monitor") + "/../_SD_SO_MON"
    // we use a URL relative to our own component; might be different if
    // extension SPR is deployed with customer namespace
});

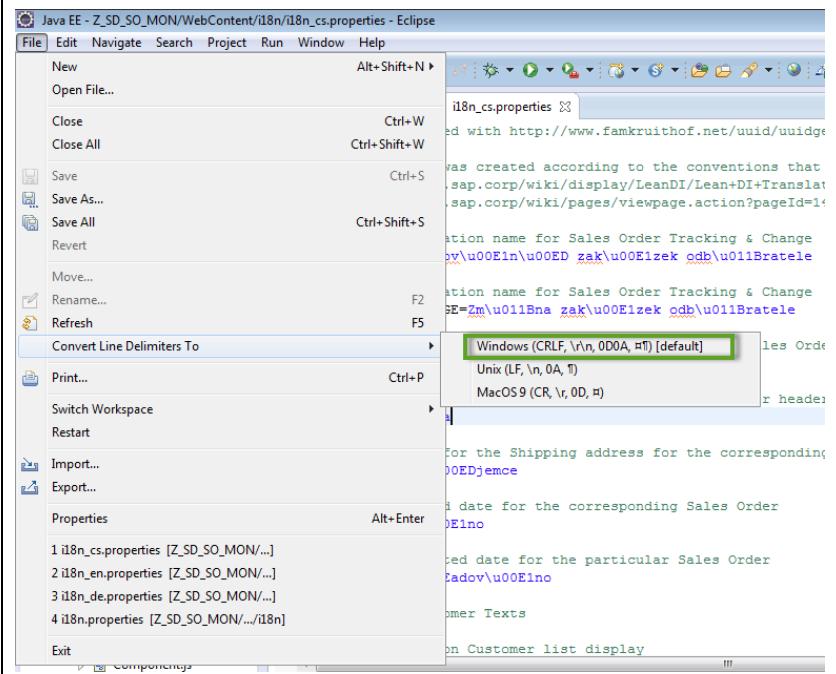
cus.sd.salesorder.monitor.Component.extend("zcus.sd.salesorder.monitor.Component", {
    metadata: {
        version: "1.0",
        config: {
            "sap.ca.i18nConfigs": {
                "bundleName": "zcus.sd.salesorder.monitor.i18n.i18n"
            },
            "sap.ca.serviceConfigs": [
                {
                    name: "ZSRA018_SO_TRACKING_SRV",
                    masterCollection: "SalesOrders",
                    serviceUrl: URI("/sap/opu/odata/sap/ZSRA018_SO_TRACKING_SRV/"),
                    directory(),
                    isDefault: true,
                    countSupported: true,
                    useBatch: true,
                    mockDataSource: jQuery.sap.getModulePath("zcus.sd.salesorder.monitor") + "/model/metadata.xml"
                }
            ],
            customizing: {
                "sap.ui.viewExtensions": [
                    "cus.sd.salesorder.monitor.view.S3": {
                        "extSOChangePaymentInfo": {
                            className: "sap.ui.core.Fragment",
                            fragmentName: "zcus.sd.salesorder.monitor.view.extSOChangePaymentInfo",
                            type: "XML"
                        }
                    }
                ]
            },
            "sap.ui.viewModifications": [
                "cus.sd.salesorder.monitor.view.S4": {
                    "SO_OH_ITEMDETAIL": {
                        "visible": false
                    }
                }
            ]
        }
    }
});

```

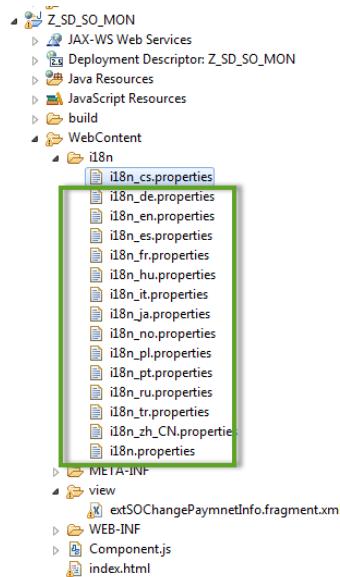
3.7.12. Now select the *i18n\_cs.properties* file by double-clicking it.



3.7.13. Select *File* → *Convert Line Delimiters To* → *Windows*, as shown in the screenshot.



3.7.14. Now repeat the steps 3.6.12 and 3.6.13 for the remaining *i18n\_xx.properties* files.



### 3.8. Testing the Application Locally before Uploading

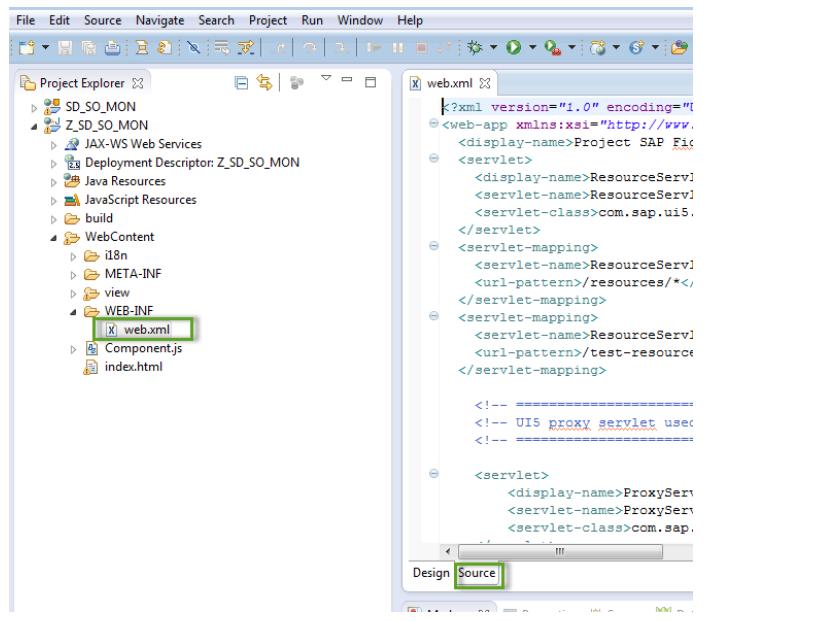
Some changes have to be done in the application to test locally.

**Note: Ensure you undo these changes before uploading the application to the SAP system.**

3.8.1. Double-click the *web.xml* file in the *Z\_SD\_SO\_MON* project as shown.

Select the *Source* tab in the *web.xml* preview.

**Note:** Your project name is *Z\_SD\_SO\_MON\_XX*.  
(Where XX is the number).



**3.8.2. Add the highlighted text to the *web.xml* file.**

Click the *Save* button.



Note: Click the icon to copy the source code.



```

<!-->
<servlet-mapping>
    <servlet-name>ProxyServlet</servlet-name>
    <url-pattern>/proxy/*</url-pattern>
</servlet-mapping>

<context-param>
    <param-name>com.sap.ui5.resource.DEV_MODE</param-name>
    <param-value>true</param-value>
</context-param>

<context-param>
    <param-name>com.sap.ui5.proxy.REMOTE_LOCATION</param-name>
    <param-value>http://cimg1.wdf.sap.corp:8101</param-value>
</context-param>

<filter>
    <display-name>CacheControlFilter</display-name>
    <filter-name>CacheControlFilter</filter-name>
    <filter-class>com.sap.ui5.resources.CacheControlFilter</filter-class>
</filter>

```

**3.8.3. Open *Component.js* and prefix the *serviceUrl* with **/approot/proxy**.**

(ie: */Z\_SD\_SO\_MON/proxy*)

**Note1:** Click the icon to copy the source code.

**Note2:** In your project, the name will be something like */Z\_SD\_SO\_MON\_XX/proxy*.

Change accordingly in the highlighted section of your source code



```

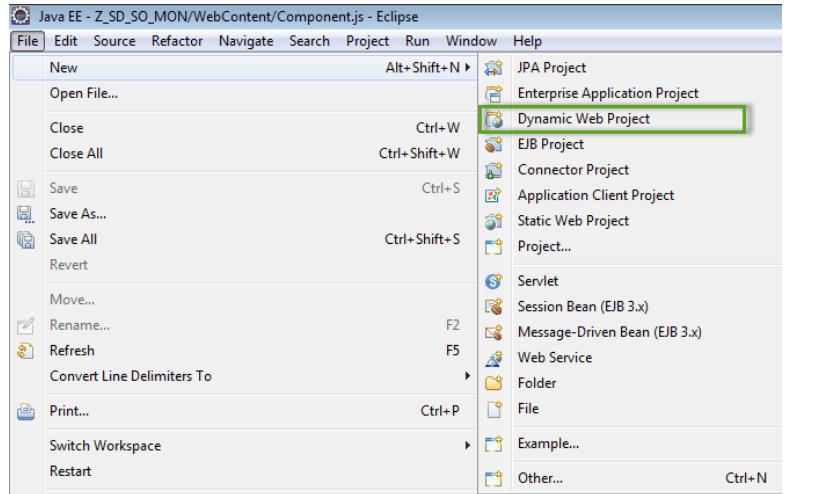
jQuery.sap.declare("zcus.sd.salesorder.monitor.Component");

// use the load function for getting the optimized preload file if present
sap.ui.component.load({
    name: 'zcus.sd.salesorder.monitor',
    url: 'sap/ui/resourceModulePath("zcus.sd.salesorder.monitor") + '/../SD_SO_MON'
        // we use a URL relative to our own component; might be different if
        // extension app is deployed with customer namespace
});
cus.sd.salesorder.monitor.Component.extend('zcus.sd.salesorder.monitor.Component', {
    metadata: {
        version: '1.0',
        config: {
            "sap.ca.i18nconfig": {
                "bundleName": "zcus.sd.salesorder.monitor.i18n.i18n"
            }
        },
        "sap.ca.serviceConfigs": [
            {
                name: "SRA018_SO_TRACKING_SRV",
                masterCollection: "SalesOrders",
                serviceUrl : URI('/Z_SD_SO_MON/proxy/sap/opu/odata/sap/ZRA018_SO_TRACKING_SRV/').directory(),
                onFailure: fme
            }
        ]
    }
});

```

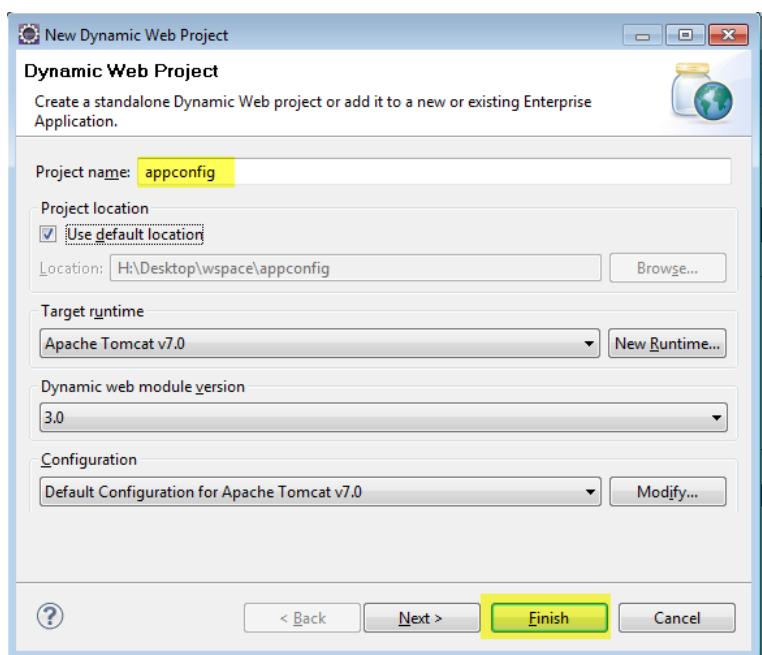
**3.8.4. It is necessary to create a configuration project to hold the application definition used by the Sandbox. To do this, create a new Web project in Eclipse.**

Select *File* → *New* → *Dynamic Web Project*.

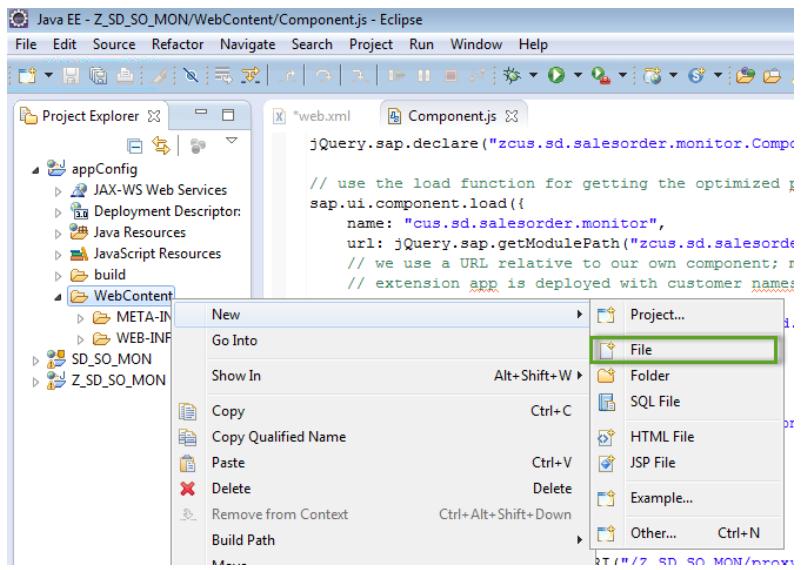


3.8.5. Use the project name appconfig.

Click *Finish*.

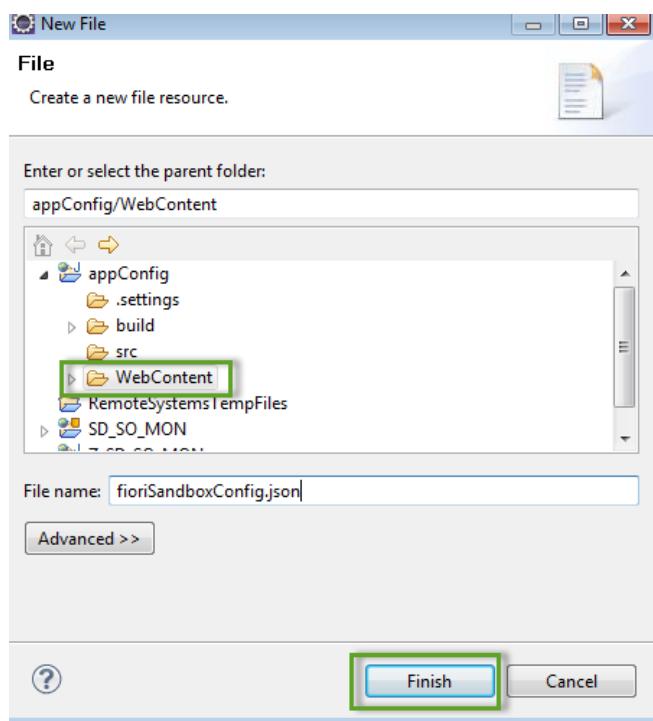


3.8.6. Within the *WebContent* folder, create a new file called *fioriSandboxConfig.json*.



3.8.7. Enter the file name as  
fioriSandboxConfig.json

Click *Finish*.



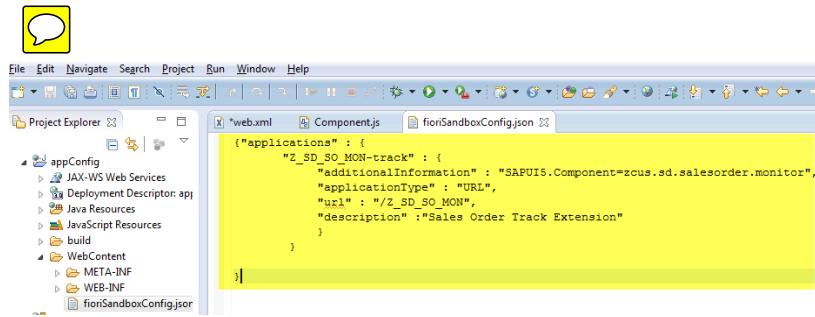
3.8.8. Add the highlighted text as shown in the screenshot.

**Z\_SD\_SO\_MON\_XX-track:** <SemanticObject>-<Action> The name of the application and the action to perform separated by a dash, which is required. (Note: where XX is the number).

- **SAPUI5.Component:** The root and namespace of the application
- **url:** The path of the app on the local server.
- **description:** The description used to display the app in the SAP Fiori launchpad

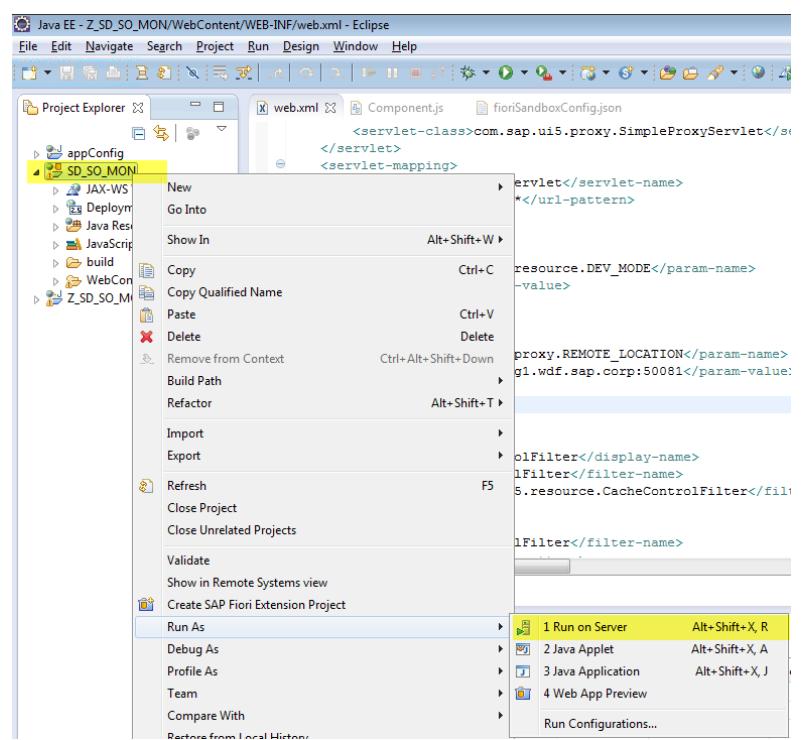
Click the *Save* button  to save the file.

**Note:** Click this icon to  copy the source code.

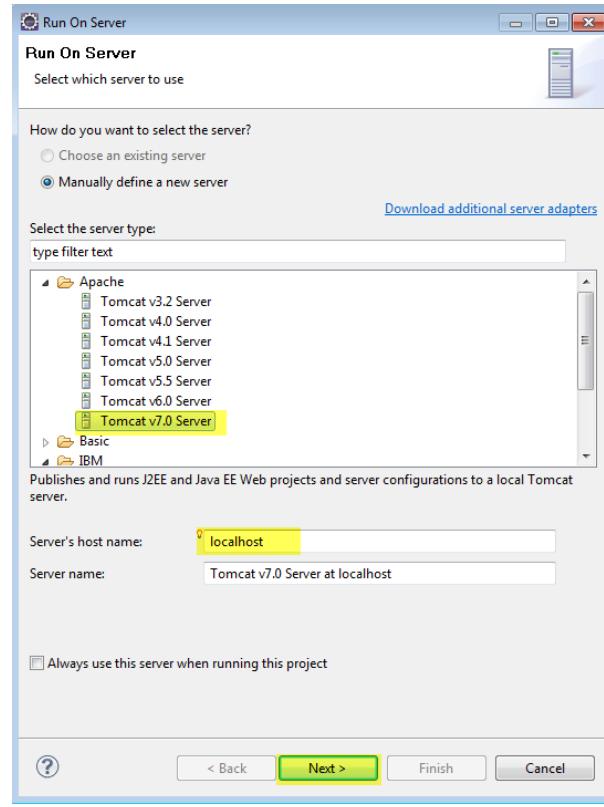


- 3.8.9. Deploy *SD\_SO\_MON*, *Z\_SD\_SO\_MON* appconfig by right-clicking on each project and selecting *Run As* → *Run on Server*.

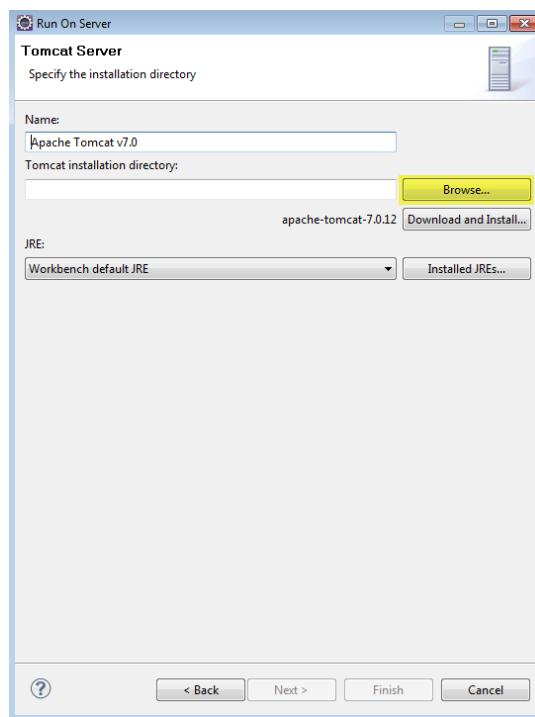
**Note:** Your project folder name will be *Z\_SD\_SO\_MON\_XX*



- 3.8.10. Select *Tomcat v7.0* and click *Next*.

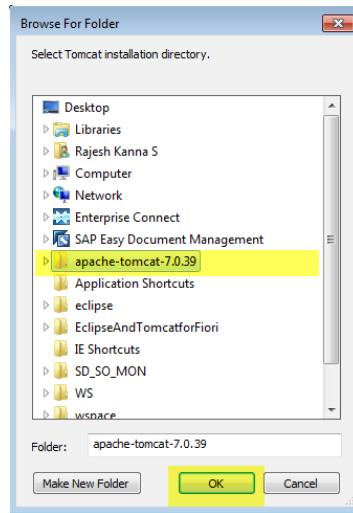


3.8.11. Click *Browse* to select the Tomcat server.

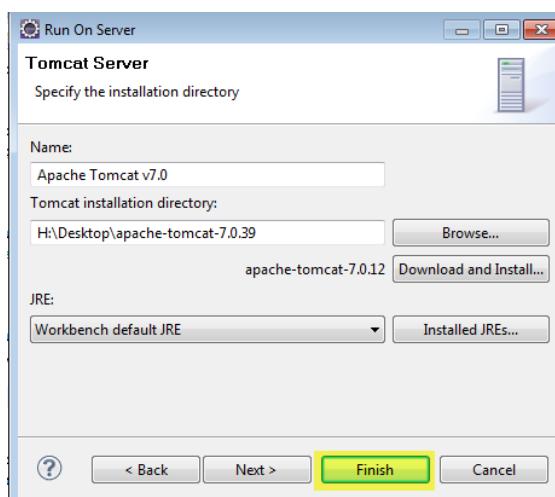


3.8.12. Select the *apache-tomcat-7* folder as we already unzipped this to the desktop (refer to point 2.3.1).

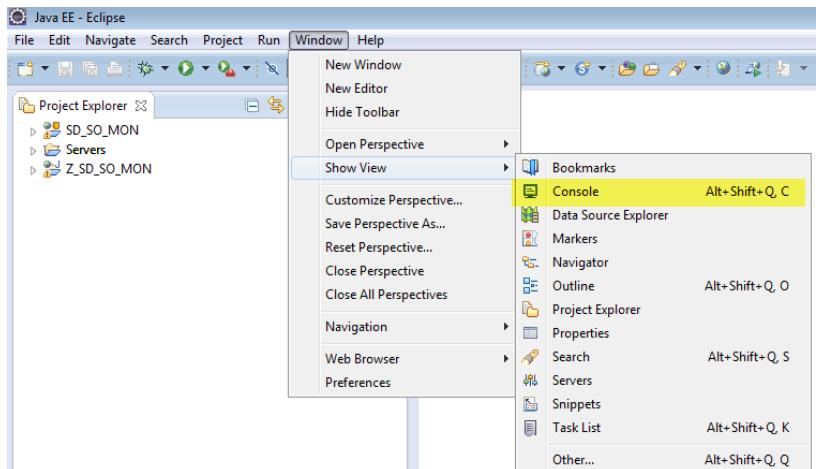
Click *OK*.



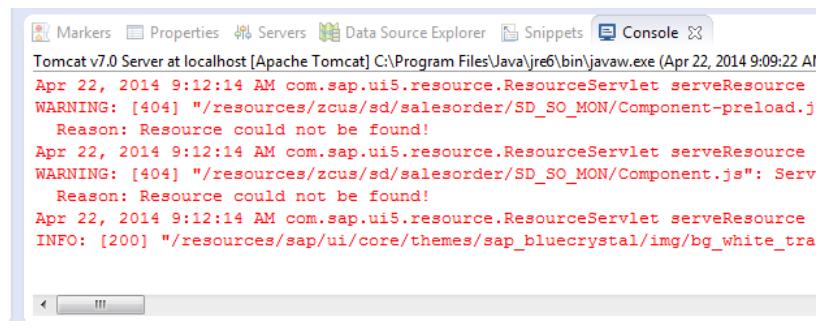
3.8.13. Click *Finish*.



3.8.14. Go to *Window* → *Show View* → *Console*.



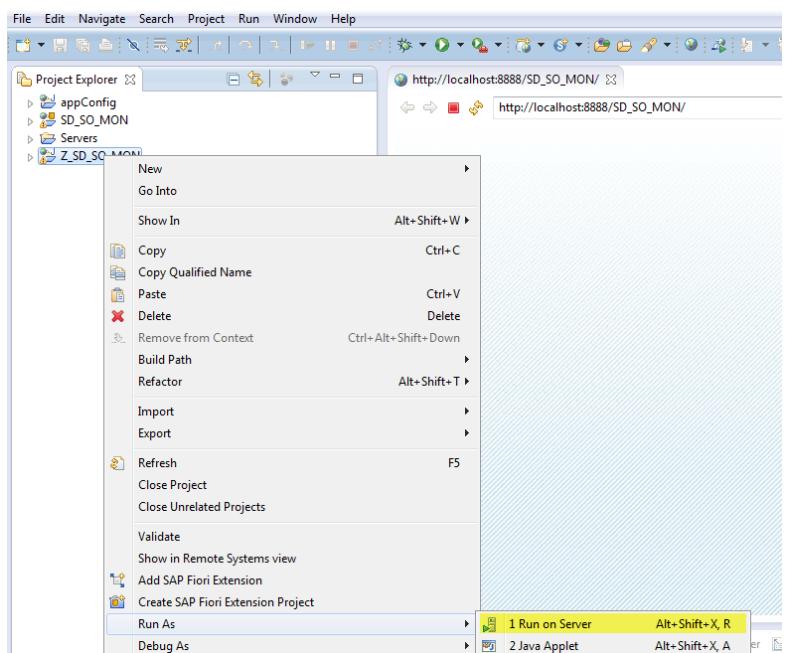
3.8.15. You can see the application-deploy messages in the console.



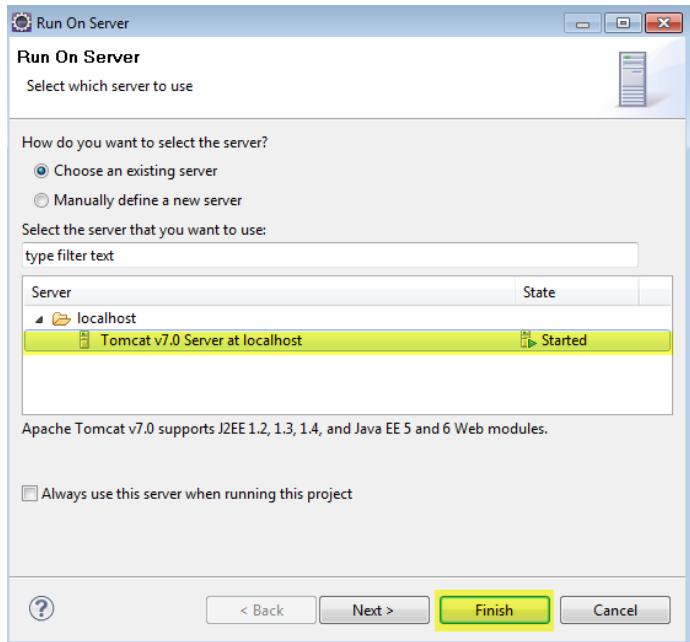
3.8.16. Now, deploy the remaining two projects.

Right-click *Z\_SD\_SO\_MON* and select *Run As* → *Run on Server*.

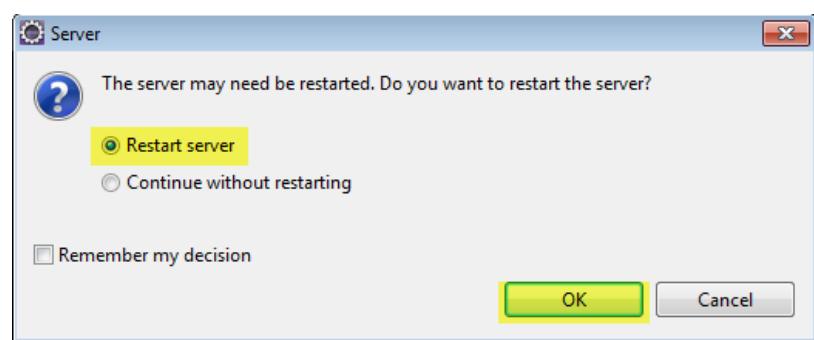
**Note:** your project name will be  
*Z\_SD\_SO\_MON\_XX*



3.8.17. Select the Tomcat server and click *Finish*.



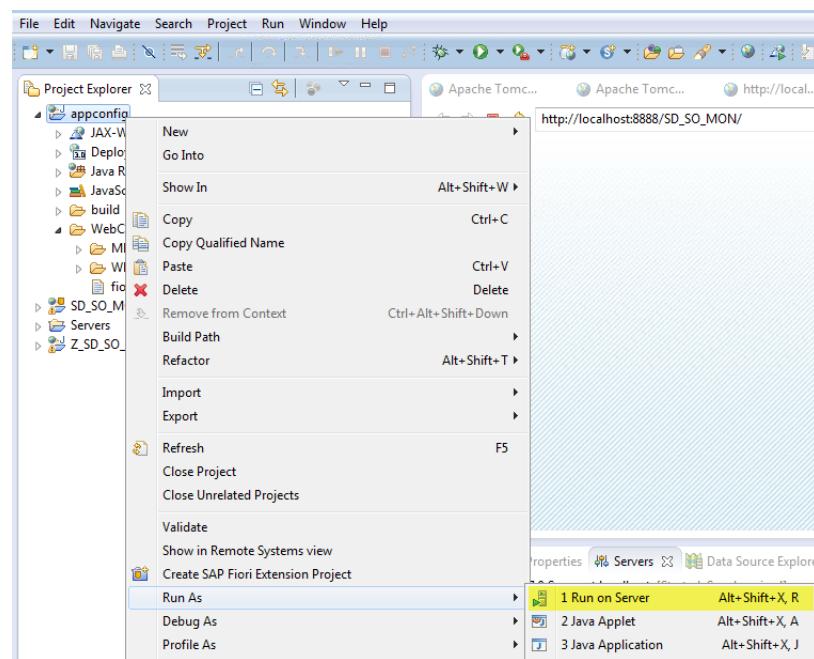
3.8.18. Select *Restart server* and click *OK*.



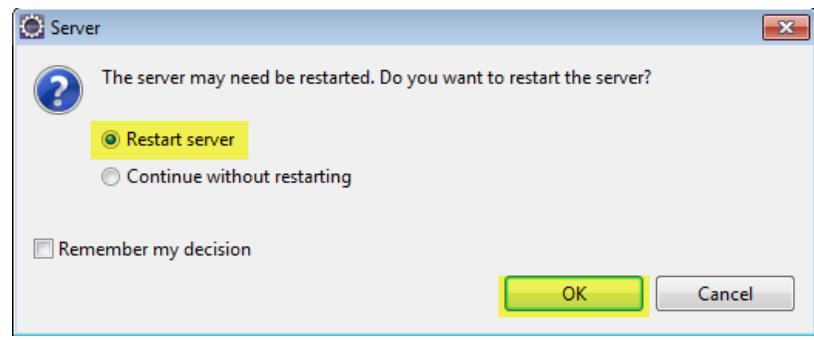
3.8.19. Right-click on *appconfig* and select *Run As* → *Run on Server*.

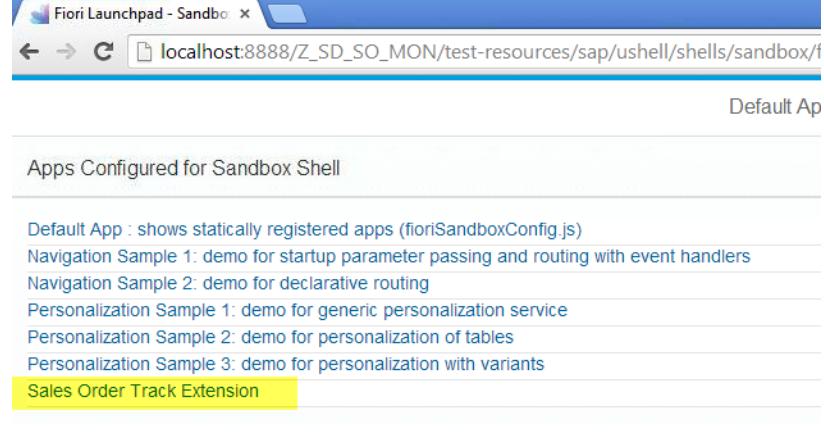
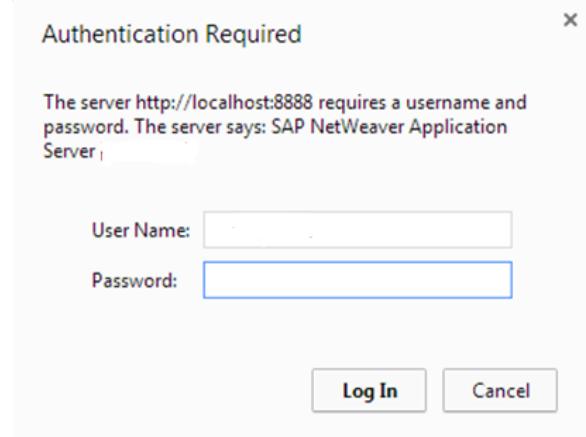
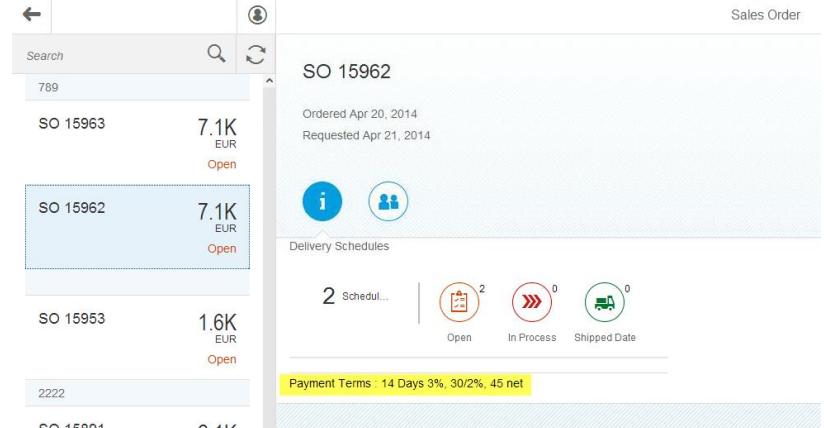
Select *Continue without restarting* in the next popup window.

Click *OK*.



3.8.20. Select *Restart server* and click *OK*.



<p>3.8.21. Now open Chrome. Enter the url   <a href="http://localhost:8888/Z_SD_SO_MON_XX/test-resources/sap/ushell/shells/sandbox/fioriSandbox.html?sap-uipreload=%20">http://localhost:8888/Z_SD_SO_MON_XX/test-resources/sap/ushell/shells/sandbox/fioriSandbox.html?sap-uipreload=%20</a>   This will load the local Launchpad to test the application. It will also list the application.   Click <i>Sales Order Track Extension</i>.   <b>Note:</b> Change the XX in the project url to your project number.</p>	
<p>3.8.22. Enter the <i>User Name</i> and <i>Password</i>.</p>	
<p>3.8.23. You can see the payment term information is displayed in the new application.</p>	

3.8.24. Click on the arrow mark in the line-item section.

Sales Order

SO 15962

Ordered Apr 20, 2014  
Requested Apr 21, 2014

Delivery Schedules

2 Scheduled | Open 2 | In Process 0 | Shipped Date 0

Payment Terms - 14 Days 3%, 30/2%, 45 net

Items (2)

Item	Quantity	Your Price	Subtotal	Status	% Shipped
10 Handle Bars-blue	10 PC	500.00 EUR	5,000.00 EUR	Open	0%
20 Handle Bars-Green	4 PC	530.00 EUR	2,120.00 EUR	Open	0%

3.8.25. We can see that the quantity and price information is hidden in the highlighted section (refer to the screenshot in point 3.1.5).

Item Detail

Search

789

SO 15963 7.1K EUR Open

SO 15962 7.1K EUR Open

SO 15953 1.6K EUR Open

2222

Information

Material No.: M-111  
Cust. Material No.:

Ship To

Shipping Address: 36 Calvinstrasse  
Berlin, 11 13467 DE  
030-8853-0

Shipping Method:

Delivery Schedules (1)

Quantity	Requested...	Promised ...	Shipped D...	Status
10 PC	Apr 20, 2014	Apr 25, 2014		Open

3.8.26. After successful testing locally, now we have to undo the changes before uploading to the SAP system.

Open *Component.js* in the *Z\_SD\_SO\_MON* project (your project name will be *Z\_SD\_SO\_MON\_XX*).

Modify the service URI as shown the highlighted text in the screenshot

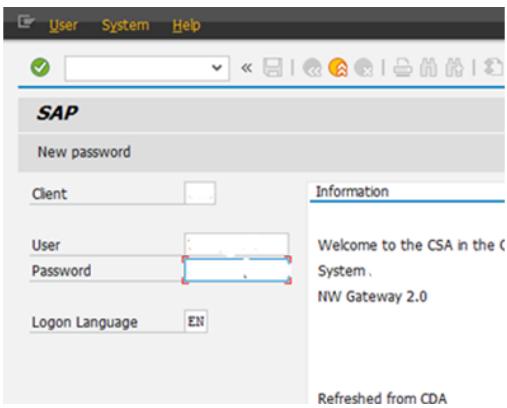
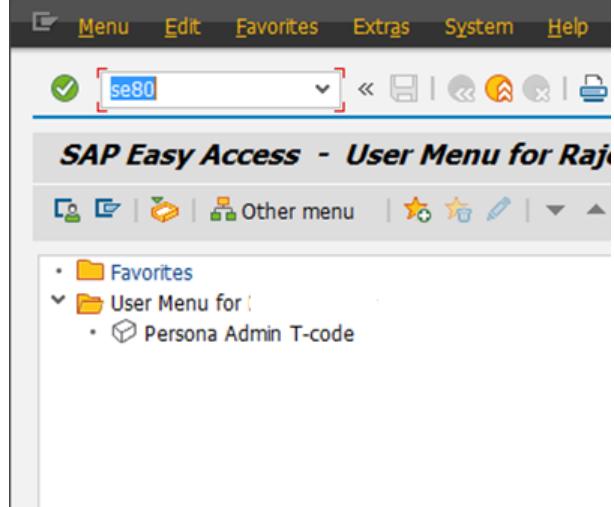
**Note:** Click on the icon to copy the source code.

If you have created your own Gateway service, please change the Gateway service name accordingly.

```

File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer Apache Tomcat... http://local... Apache Tomcat... http://local...
Z_SD_SO_MON_Z_SD_SO_MON_JAX-WS Web Service Deployment Descrij Java Resources JavaScript Resource build WebContent i18n META-INF view extSOChang WEB-INF Component.js index.html
sap.ca.serviceConfigs": [{"name: "ZRA018_SO_TRACKING_SRV", masterCollection : "SalesOrders", serviceUrl : URI("/sap/cuiodata/sap/ZRRA018_SO_TRACKING_SRV/"), directory(), isDefault : true, countSupported : true, useBatch : true, mockDataSource : jQuery.sap.getModulePath("zcus.sd.salesorder.monitor")}], customizing: {"sap.ui.viewExtensions": {"cus.sd.salesorder.monitor.view.S3": {"extSOChangePaymentInfo": {"className: "sap.ui.core.Fragment", fragmentName: "zcus.sd.salesorder.monitor.view.extSOChangePaymentInfo", "owner": null}}}}
```

### 3.9. Upload the Application into the SAP System

What to do	What to say
<p>3.9.1. Login to the SAP Gateway system with the appropriate user and password.</p>	
<p>3.9.2. Before <b>uploading/re-uploading</b> the application, you have to make sure that there is no application with the same BSP name in the system.      You are going to upload the application with the name <i>Z_SD_SO_MON</i>.      So first you have make sure that there is no BSP application in the system with the name <i>Z_SD_SO_MON</i>.      To do this run transaction <i>SE80</i>.</p> <p><b>Note:</b> In your case the project name will be <i>Z_SD_SO_MON_XX</i> (where XX is your appended number).</p>	

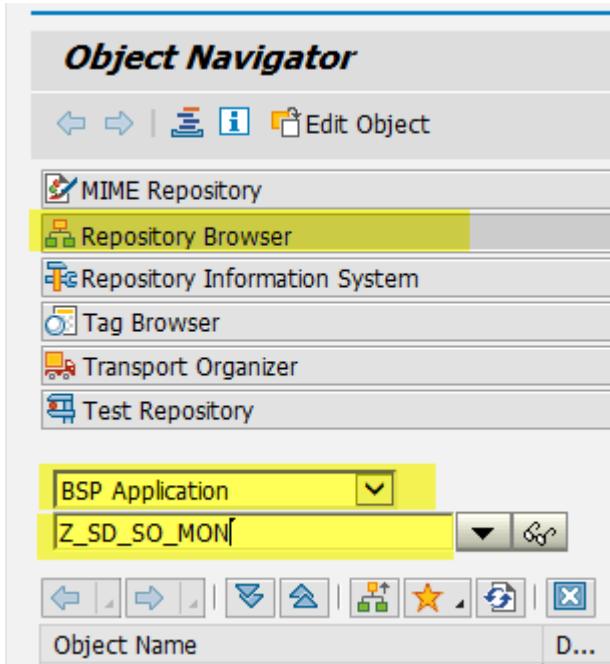
3.9.3. Select *Repository Browser*.

Choose *BSP Application* on the dropdown list.

Enter the application name as  
Z\_SD\_SO\_MON\_XX (name of the BSP  
application you are going to upload).

Press *Enter*.

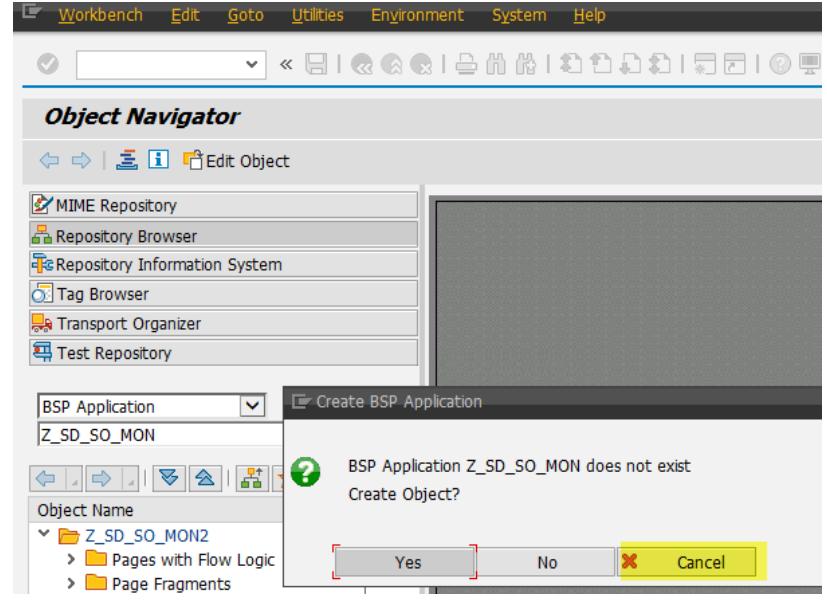
**Note:** In your case it will be  
Z\_SD\_SO\_MON\_XX



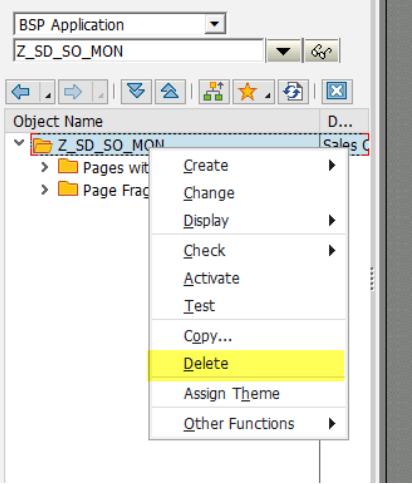
3.9.4. If the application exists, it will load the application. Otherwise, popup appears saying there is no BSP application.

If you want to upload an application with the same name, you have to delete the existing application.

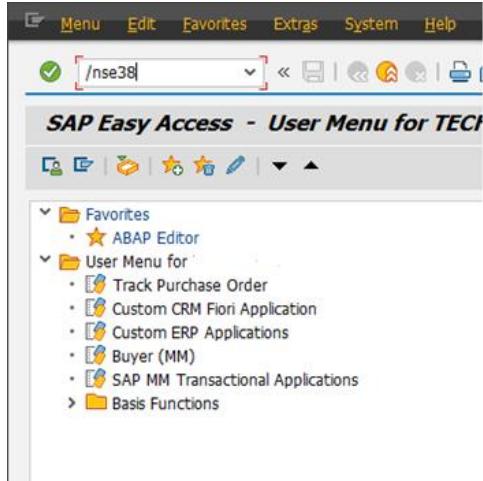
If it does not exist, click *Cancel*.



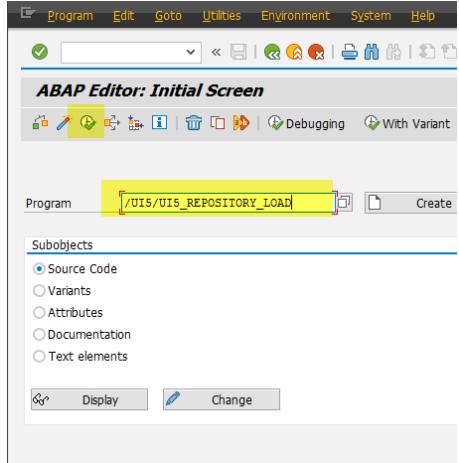
- 3.9.5. To delete an application, right-click the application and choose *Delete*.

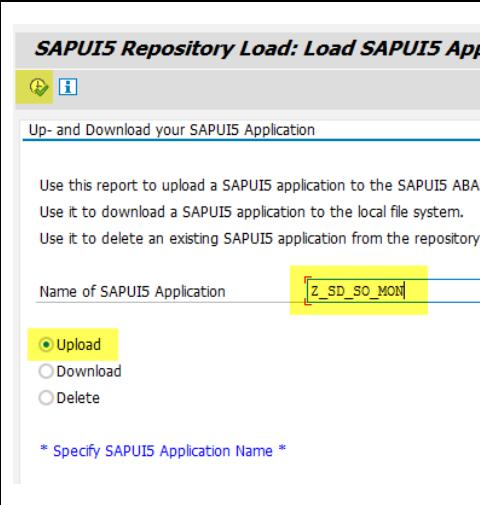
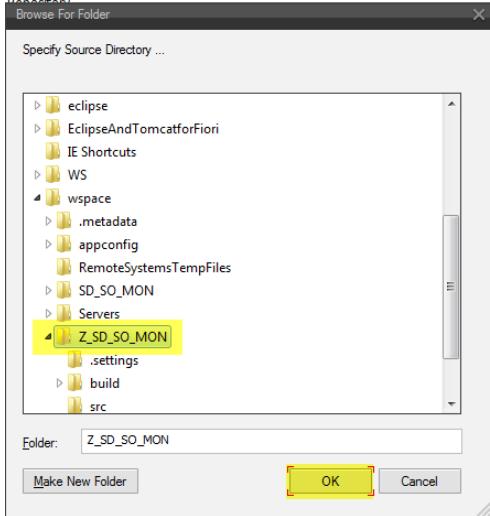
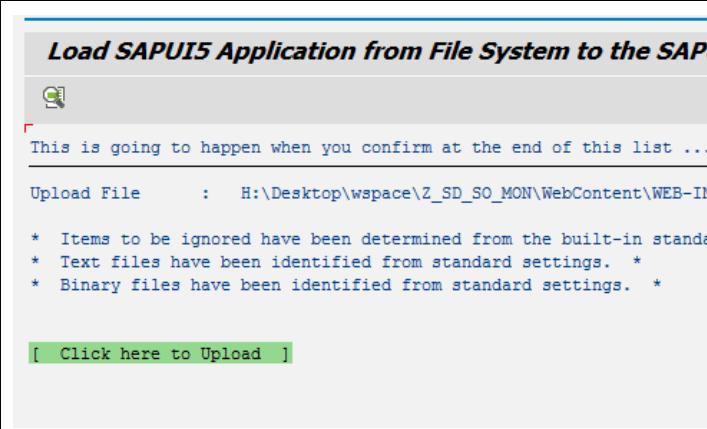


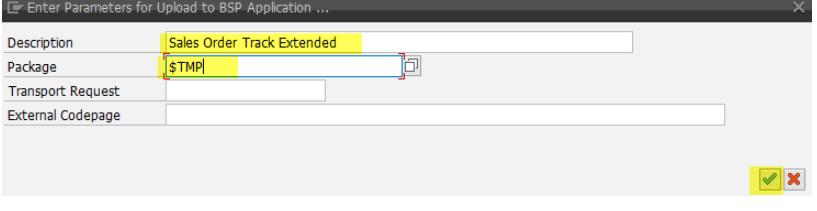
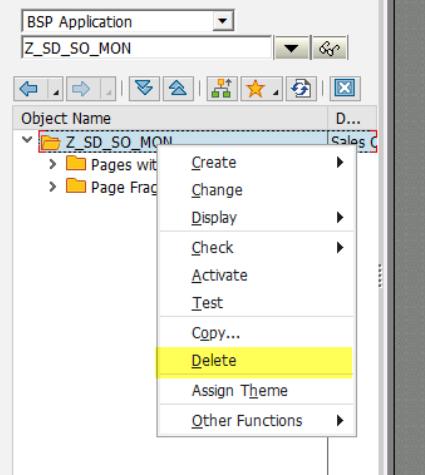
- 3.9.6. Run transaction SE38.



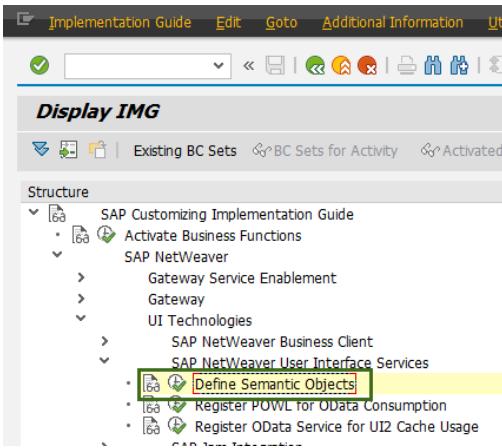
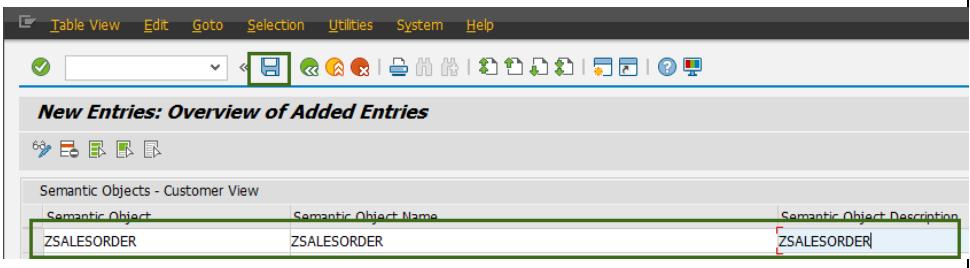
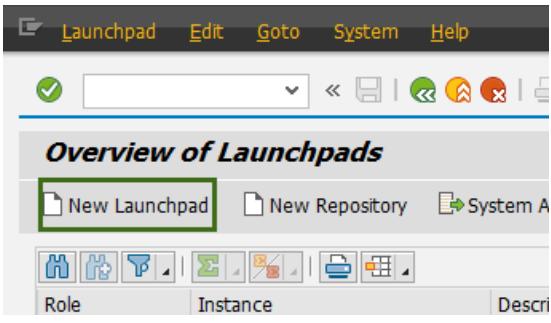
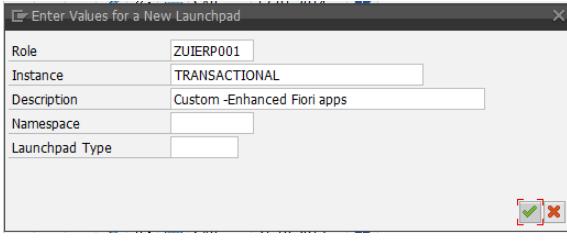
- 3.9.7. In the *Program* field, enter /UI5/UI5\_REPOSITORY\_LOAD and click the *Execute* button.

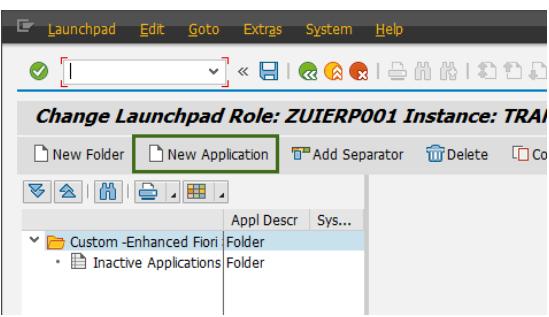


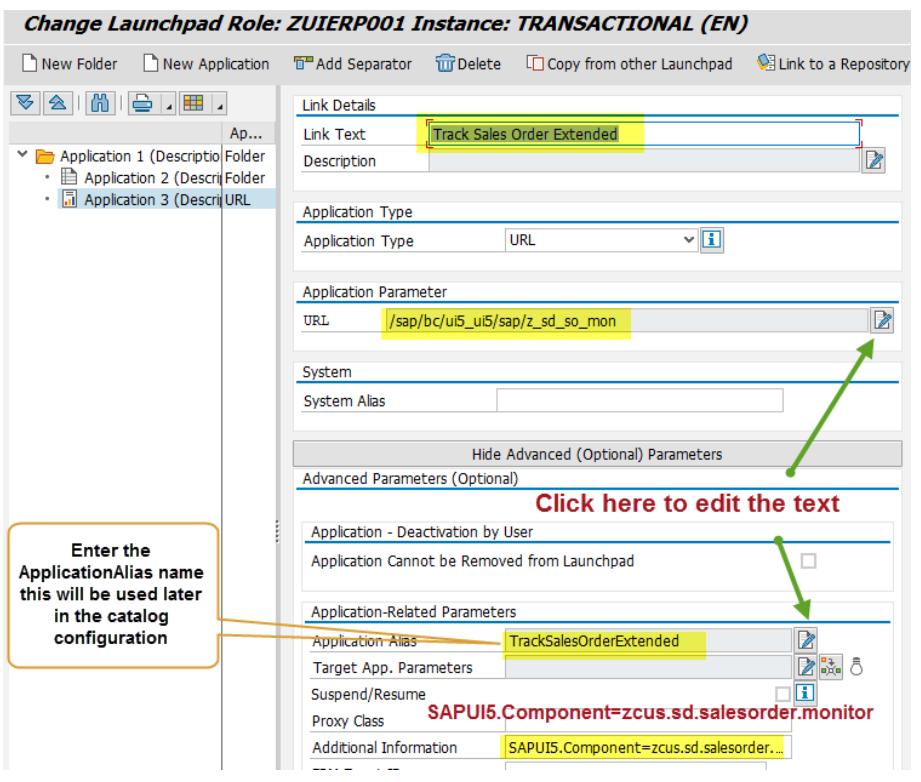
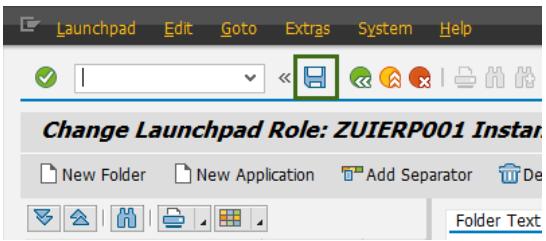
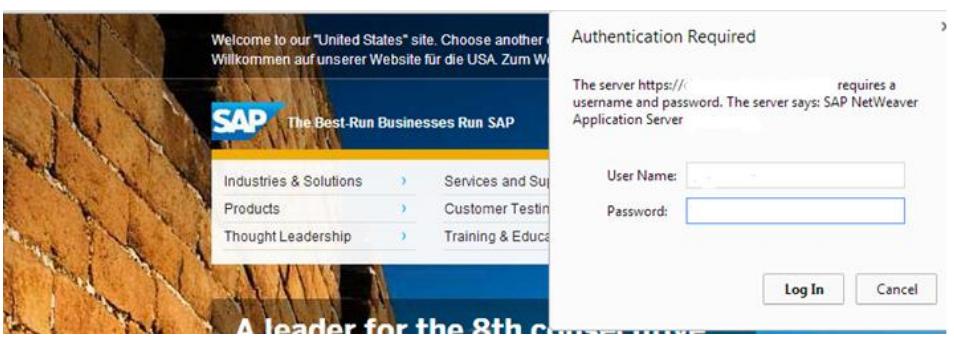
<p>3.9.8. Enter the application name as Z_SD_SO_MON_XX, and select the <i>Upload</i> radio button.</p> <p><b>Note:</b> In your case the project name will be Z_SD_SO_MON (where XX is the number you chose).</p>	
<p>3.9.9. Select the Z_SD_SO_MON project folder inside the workspace.</p> <p>Click <i>OK</i>.</p>	
<p>3.9.10. Scroll down to the end of the page.</p> <p>Select <i>Click here to Upload</i>.</p>	

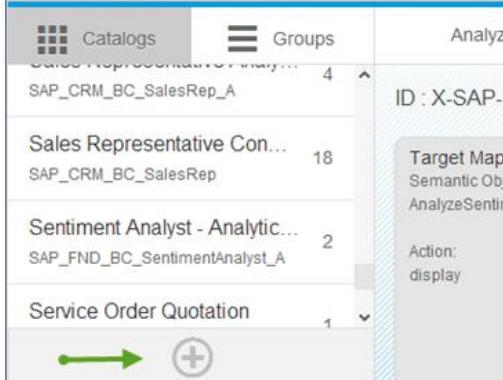
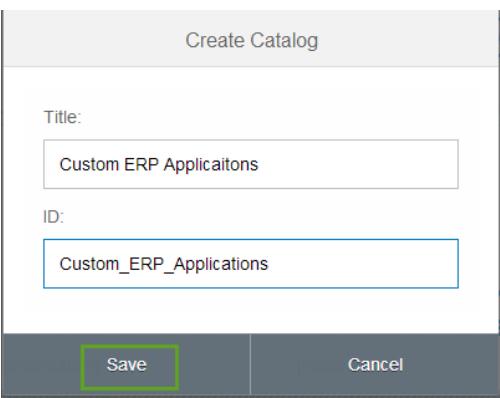
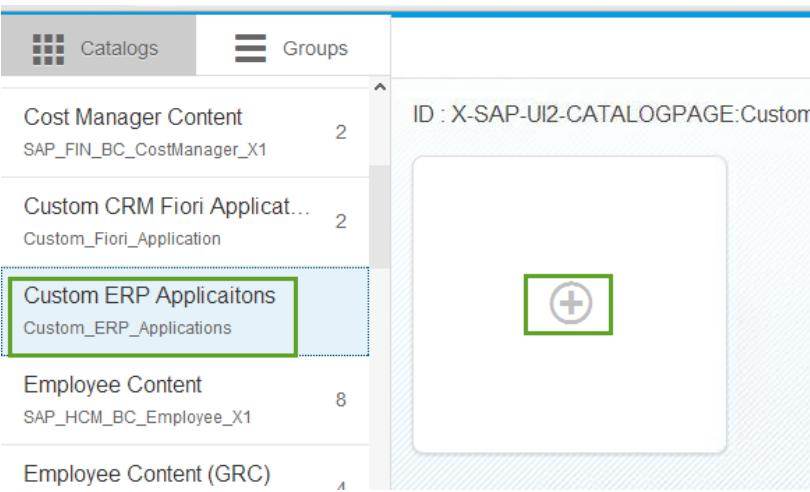
<p>3.9.11. Enter the description and package as \$TMP Click on the green tick mark.</p>	
<p>3.9.12. The success message appears, as shown in the screenshot.  Any message or screen other than this means there is an error somewhere.</p>	<p>&gt; FOR OPERATION YOU MAY NEED AUTHORIZATION FOR THE FOLLOWING AUTHORITY S_DEVELOP, S_ICF ADM, S_TCODE, S_TRANSPRT, S_CTS ADM Check the security guide.</p> <p> The object will be created in the original language English (EN)</p>
<p>3.9.13. If you want to re-upload the same application, you need to delete the existing application.  Refer to point 3.9.5 for more information.</p>	

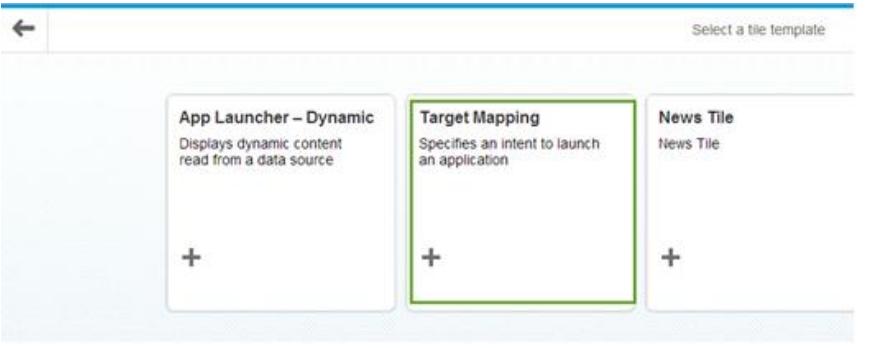
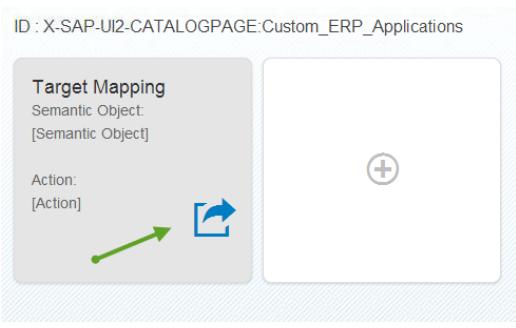
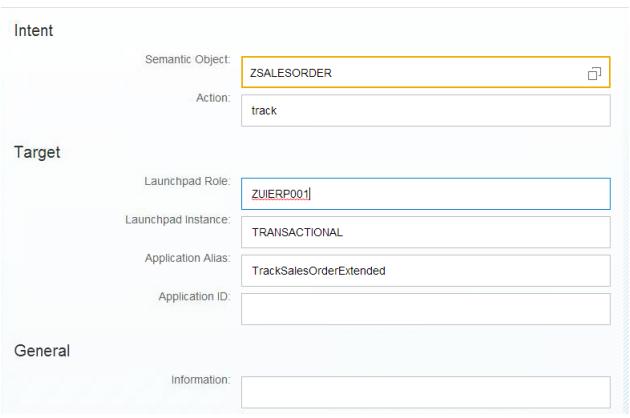
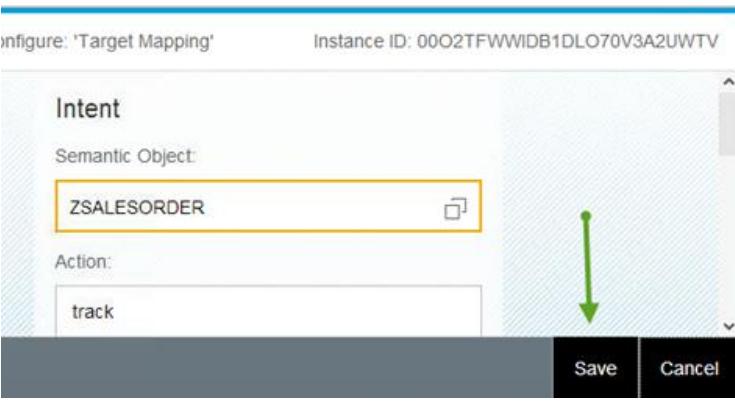
### 3.10. Configuring the New Application in the SAP Fiori Launchpad

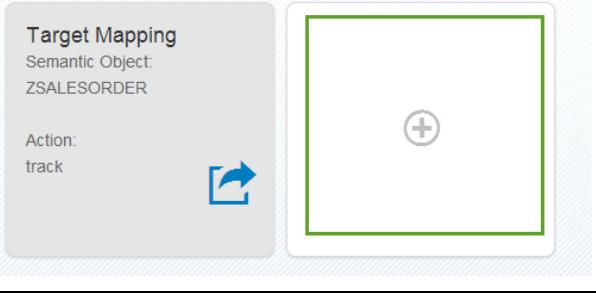
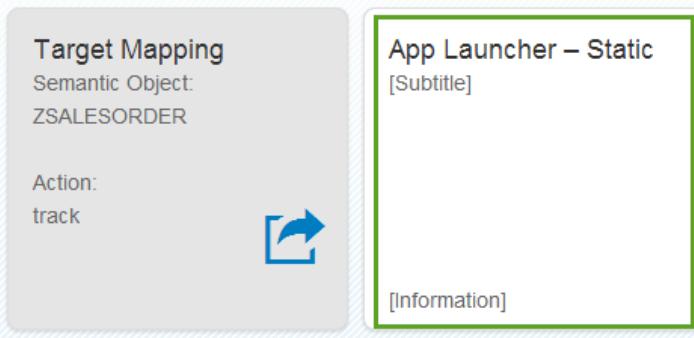
<p>3.10.1. Run transaction SPRO.</p> <p>Click <i>Define Semantic Objects</i> as shown in the screenshot.</p>							
<p>3.10.2. Enter the semantic object, name, and description.</p> <p>Click <i>Save</i>.</p> <p>Note: Please enter the semantic object name as ZSALESORDER_XX (where XX is the number you chose).</p>	 <table border="1" data-bbox="584 988 1510 1045"> <thead> <tr> <th>Semantic Object</th> <th>Semantic Object Name</th> <th>Semantic Object Description</th> </tr> </thead> <tbody> <tr> <td>ZSALESORDER</td> <td>ZSALESORDER</td> <td>ZSALESORDER</td> </tr> </tbody> </table>	Semantic Object	Semantic Object Name	Semantic Object Description	ZSALESORDER	ZSALESORDER	ZSALESORDER
Semantic Object	Semantic Object Name	Semantic Object Description					
ZSALESORDER	ZSALESORDER	ZSALESORDER					
<p>3.10.3. Run transaction LPD_CUST.</p> <p>Click <i>New Launchpad</i>.</p>							
<p>3.10.4. Enter the new Launchpad details and click the green tick mark.</p> <p>Note down this information as it will be used later in the catalog configuration.</p> <p>On the popup window, click <i>Yes</i>.</p>							

<p><b>Note:</b> The launch pad ZUIERP001 is already created in the system.</p> <p>If you want to create your own Launchpad, please keep the Launchpad role as <i>ZUIERP0XX</i> (where XX is the number you are using in the scenario).</p> <p>It is not necessary that you have to create a new Launchpad for every application. One Launchpad can hold a number of applications.</p> <p>You can use the existing Launchpad or you can create your own.</p>	
3.10.5. Click <i>New Application</i> .	

<p>3.10.6. Enter the application details as shown in the screenshot.</p> <p><b>Note:</b> Enter the project name of yours such as Z_SD_SO_MON_XX (where XX is the number) in the URL section.</p>	
<p>3.10.7. Click Save and then the Back button.</p>	
<p>3.10.8. Go to the Fiori Admin page. Login with User Name and Password.  Replace &lt;your_ABAP_URL&gt; with http://&lt;ABAP server&gt;:&lt;port&gt;</p>	<p>Admin page url -  <a href="http://&lt;your_ABAP_URL&gt;/sap/bc/ui5_ui5/sap/arsvc_upb_admn/main.html?scope=CUST">http://&lt;your_ABAP_URL&gt;/sap/bc/ui5_ui5/sap/arsvc_upb_admn/main.html?scope=CUST</a></p> 

<p>3.10.9. Click on the + icon to create a new custom catalog.</p>	
<p>3.10.10.</p> <p>Enter the catalog <i>Title</i> and <i>ID</i>. Click <i>Save</i>.</p> <p><b>Note:</b> The catalog shown in the screenshot already exists in the system.</p> <p>Either you can use the existing catalog or you can create your own catalog.</p> <p>One catalog can have any number of applications.</p>	
<p>3.10.11.</p> <p>Select the catalog you just created.</p> <p>Click the + icon on the newly created custom catalog.</p>	

3.10.12. Select <i>Target Mapping</i> in the list of tile templates.	
3.10.13. Select <i>Target Mapping</i> again to configure the new application.	
3.10.14. Enter the <i>Semantic Object</i> , <i>Launchpad Role</i> , <i>Launchpad instance</i> , and the <i>Application Alias</i> as we created before (Refer to points 3.10.2. and 3.10.4.).	
3.10.15. Click <i>Save</i> .	

3.10.16. Click the + icon next to <i>Target Mapping</i> .	<p>ID : X-SAP-UI2-CATALOGPAGE:Custom_ERP_Applications</p> 																				
3.10.17. Click <i>App Launcher - Static</i> .																					
3.10.18. Select <i>App Launcher - Static</i> again.	<p>ID : X-SAP-UI2-CATALOGPAGE:Custom_ERP_Applications</p> 																				
<p>3.10.19. Enter the details as shown in the screenshot.  Click <i>Save</i> in the right-bottom corner of the page.</p> <p><b>Note:</b> Refer to point 3.10.2 for the semantic object name you have created.</p>	<p>General</p> <table border="1"> <tr> <td>Title:</td> <td>Sales order Extended</td> </tr> <tr> <td>Subtitle:</td> <td></td> </tr> <tr> <td>Keywords:</td> <td>Sales order</td> </tr> <tr> <td>Icon:</td> <td>sap-icon://Fiori/F0020</td> </tr> <tr> <td>Information:</td> <td></td> </tr> </table> <p>Navigation</p> <table border="1"> <tr> <td>Use semantic object navigation:</td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>Semantic Object:</td> <td>ZSALESORDER</td> </tr> <tr> <td>Action:</td> <td>track</td> </tr> <tr> <td>Parameters:</td> <td></td> </tr> <tr> <td>Target URL:</td> <td></td> </tr> </table>	Title:	Sales order Extended	Subtitle:		Keywords:	Sales order	Icon:	sap-icon://Fiori/F0020	Information:		Use semantic object navigation:	<input checked="" type="checkbox"/>	Semantic Object:	ZSALESORDER	Action:	track	Parameters:		Target URL:	
Title:	Sales order Extended																				
Subtitle:																					
Keywords:	Sales order																				
Icon:	sap-icon://Fiori/F0020																				
Information:																					
Use semantic object navigation:	<input checked="" type="checkbox"/>																				
Semantic Object:	ZSALESORDER																				
Action:	track																				
Parameters:																					
Target URL:																					

### 3.10.20.

Now we need to assign the catalog to profiles.

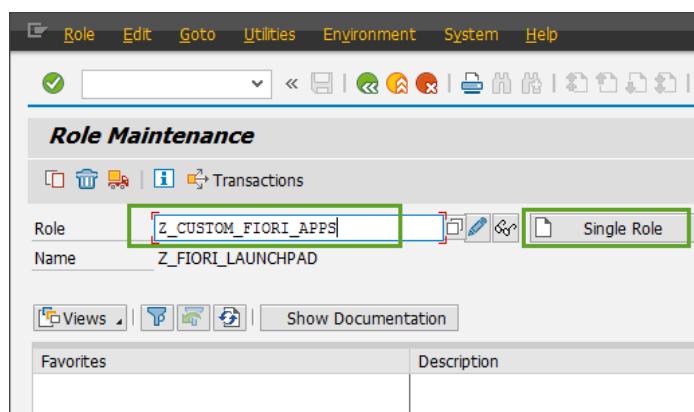
To do this login back to the Gateway system.

Run transaction PFCG.  
Enter the role name and click *Single Role*.

**Note:** The role shown in the screenshot is already available in the system.

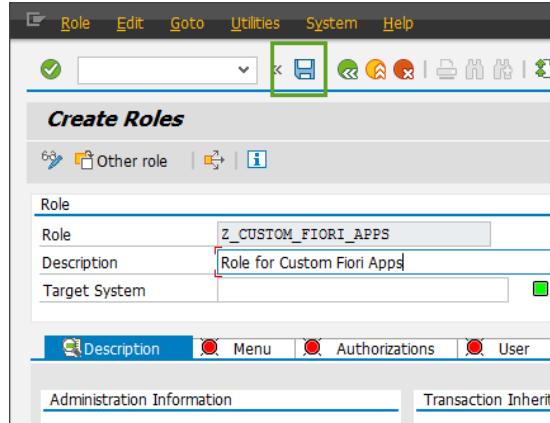
You can either use existing Role or you can create your own Role such as *Z\_CUSTOM\_FIORI\_APPS\_XX* (Where XX is the number.)

One Role can have n number catalog profiles.



### 3.10.21.

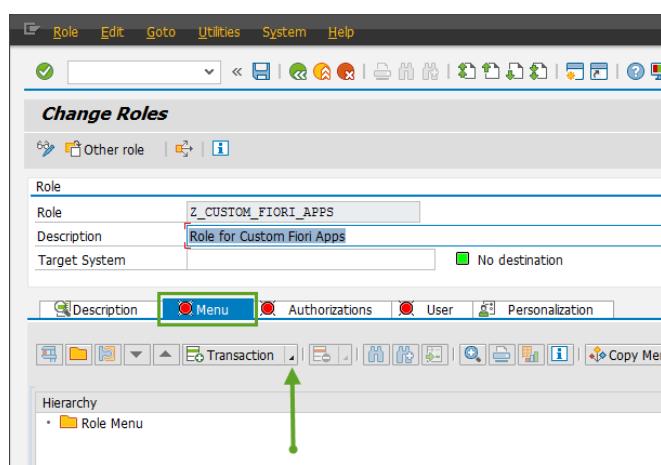
Enter the description and click *Save*.



### 3.10.22.

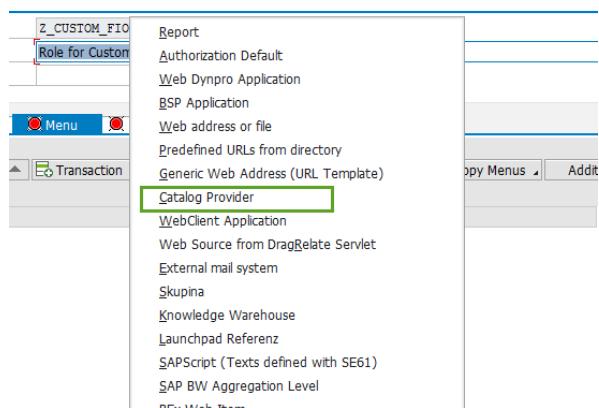
Select the *Menu* tab.

Click the arrow mark in the *Transaction* button.



### 3.10.23.

Select *Catalog Provider* from the menu.

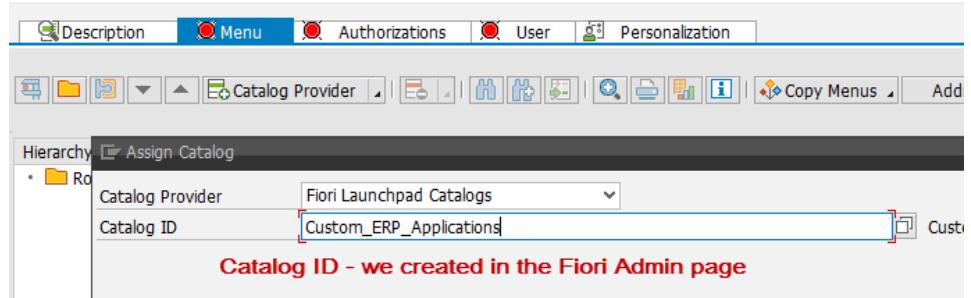


### 3.10.24.

Enter the *Catalog ID* we created in the Fiori Admin page.

Refer to point 3.10.10

**Note:** The catalog *Custom\_ERP\_Applications* is already added to the role, *Z\_CUSTOM\_FIORI\_APPS*. You can either add your catalog to the existing role or you can add it to your new role, if you have created one.



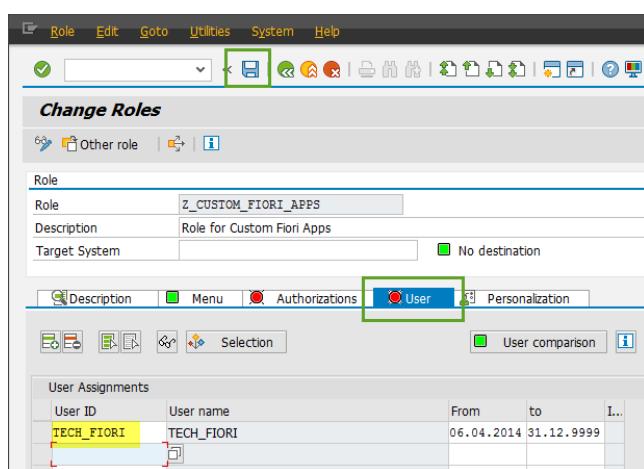
3.10.25.

Select the *User* tab.

Enter the list of users to be added to the Fiori catalog.

Click *Save*.

**Note:** This custom application will be visible only if we assign the users to this profile.



3.10.26.

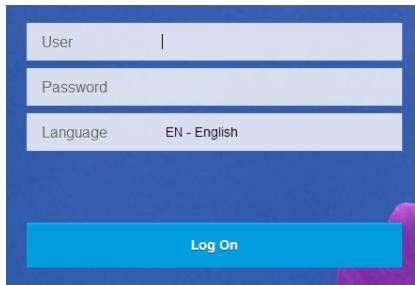
Login to the Fiori launchpad.

Login with the configured user.

Replace <your\_ABAP\_URL> with `http://<ABAP server>:<port>`

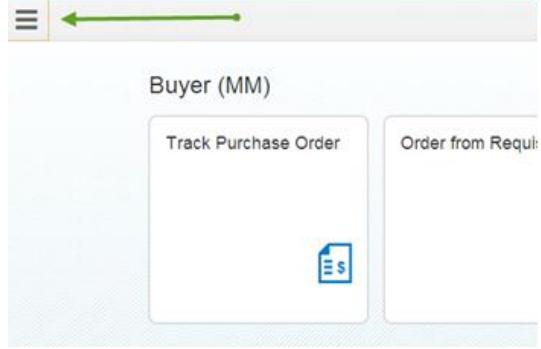
**Fiori Launchpad URL**

`http://< your_ABAP_URL >/sap/bc/ui5_ui5/ui2/ushell/shells/abap/FioriLaunchpad.html`



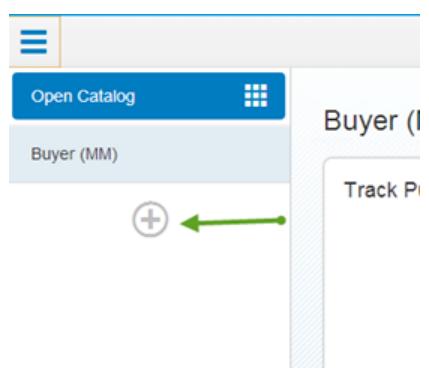
3.10.27.

Click the *Menu* icon.



3.10.28.

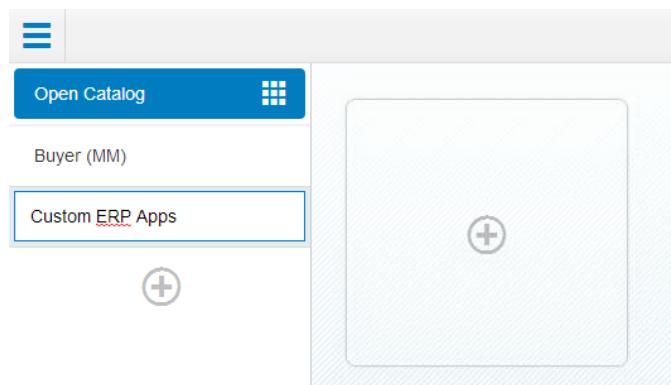
Click the *Add* icon to add the group.



3.10.29.

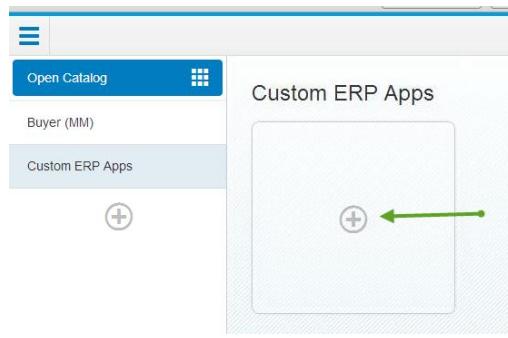
Type the group name and press Enter.

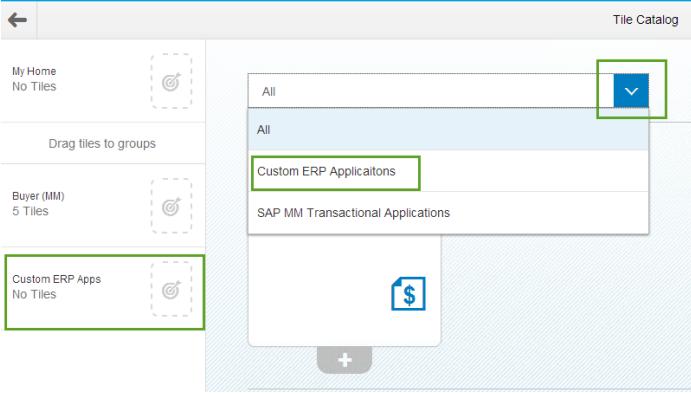
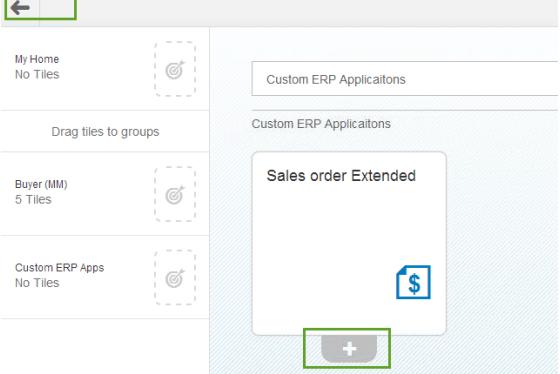
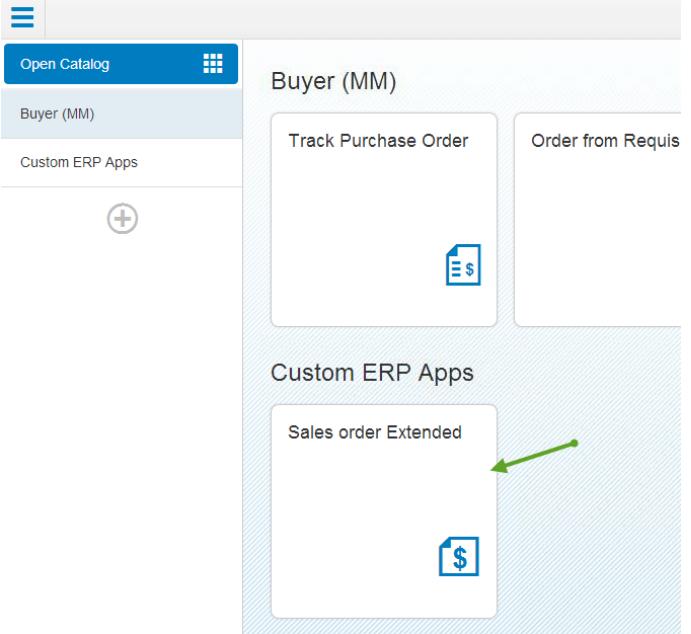
**Note:** You can either use an existing group or you can create your own group

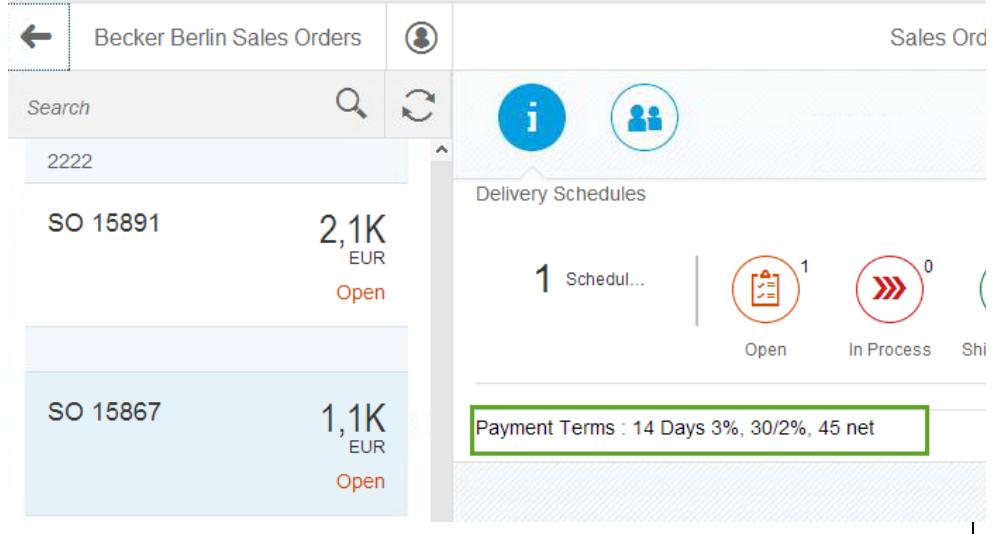
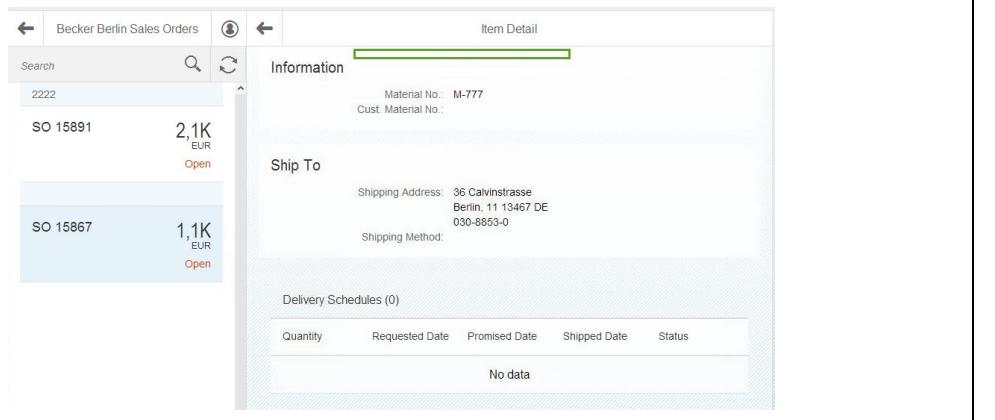


3.10.30.

Click the *Add* icon to add applications to the group.



<p>3.10.31.</p> <p>Select the <i>Custom ERP Apps</i> group in left side menu.</p> <p>Select <i>Custom ERP Applications (Catalog)</i> from the dropdown menu.</p> <p><b>Note:</b> Select the catalog you have created.</p>	
<p>3.10.32.</p> <p>You can see the list of applications added to this catalog.</p> <p>Click the + icon on the <i>Sales order Extended</i> tile.</p> <p>Click the <i>Back</i> icon to go back.</p>	
<p>3.10.33.</p> <p>Now the user is configured for the application <i>Sales order Extended</i>.</p> <p>Click the <i>Sales order Extended</i> application to open it.</p>	

3.10.34.	<p><b>Output:</b> Payment terms is displayed (Refer to Point 3.1.3.)</p> 
3.10.35.	<p><b>Output:</b> Item detail Header is invisible. (Refer the Point 3.1.5)</p> 

## © 2014 SAP AG OR AN SAP AFFILIATE COMPANY. ALL RIGHTS RESERVED.

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.

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

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

Bluetooth is a registered trademark of Bluetooth SIG Inc.

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

Computop is a registered trademark of Computop Wirtschaftsinformatik GmbH.

Edgar Online is a registered trademark of EDGAR Online Inc., an R.R. Donnelley & Sons Company.

Facebook, the Facebook and F logo, FB, Face, Poke, Wall, and 32665 are trademarks of Facebook.

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.

HP is a registered trademark of the Hewlett-Packard Development Company L.P.

HTML, XML, XHTML, and W3C are trademarks, registered trademarks, or claimed as generic terms by the Massachusetts Institute of Technology (MIT), European Research Consortium for Informatics and Mathematics (ERCIM), or Keio University.

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.

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

INTERMEC is a registered trademark of Intermec Technologies Corporation.

IOS is a registered trademark of Cisco Systems Inc.

The Klout name and logos are trademarks of Klout Inc.

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

Motorola is a registered trademark of Motorola Trademark Holdings LLC.

Mozilla and Firefox and their logos are registered trademarks of the Mozilla Foundation.

Novell and SUSE Linux Enterprise Server are registered trademarks of Novell Inc.

OpenText is a registered trademark of OpenText Corporation.

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

QR Code is a registered trademark of Denso Wave Incorporated.

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

SAVO is a registered trademark of The Savo Group Ltd.

The Skype name is a trademark of Skype or related entities.

Twitter and Tweet are trademarks or registered trademarks of Twitter.

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

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

SAP, R/3, ABAP, BAPI, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP BusinessObjects Explorer, StreamWork, SAP HANA, the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, Sybase, Adaptive Server, Adaptive Server Enterprise, iAnywhere, Sybase 365, SQL Anywhere, Crossgate, B2B 360° and B2B 360° Services, m@gic EDDY, Ariba, the Ariba logo, Quadrem, b-process, Ariba Discovery, SuccessFactors, Execution is the Difference, BizX Mobile Touchbase, It's time to love work again, SuccessFactors Jam and BadAss SaaS, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany or an SAP affiliate 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.