

# Integrating ABAP Web Dynpro and R/3 Transaction into Composite Applications Built with SAP Guided Procedures

## Applies to:

SAP NetWeaver 04s SP08

## Summary

While developing composite applications reuse of existing UI is critical as it drastically reduces the time taken to create such an application and it also enriches its functionality. This article explains how to incorporate already existing ABAP WebDynpro UI into a composite application that is built using SAP Guided Procedures and SAP Visual Composer. This article tries to explain it with the help of an example business scenario. Later the article also explains how a similar approach can be taken to integrate R/3 transactions in Guided Procedure processes. Note that the method described here can be achieved using the GP callable object types but there are subtle advantages as mentioned in [Point to note](#) section.

**Author:** Sijesh Manohar V

**Company:** SAP Labs India

**Created on:** 18th October 2006

## Author Bio



Sijesh Manohar V is a Development Manager at SAP Labs India. He has been working recently in development of composite application using SAP Netweaver tools and technology like CAF, Guided procedures, Visual Composer. He joined SAP Labs in 1998 as a developer in ABAP. Additionally, he has a very strong background in the industry solution IS-OIL-Downstream and Upstream (PRA) and also other Netweaver components - XI, Portal and Master Data Management.

## Table of Contents

Author Bio .....	1
Example Scenario – Quotation Approval.....	3
Actions Performed by the Sales Representative .....	3
1.    Select the Quotation from the ERP system.....	3
2.    Add Comments to the Approver.....	3
3.    Edit the Quotation Letter. ....	4
4.    Preview the Created non editable Quotation letter. ....	5
Enhancement Required .....	6
Adding the Lean Order ABAP WD UI in the Edit Quotation Step.....	6
Step 1: Create iView .....	6
1.    Identify the ABAP Webdynpro application. ....	6
2.    Create a SAP WebDynpro iView in Portal Content Studio. ....	7
3.    Create the Visual Composer iView. ....	9
Step 2: Create the Callable Object .....	11
4.    Add the Visual Composer iView to the Portal role. ....	11
5.    Create a new Callable Object (CO).....	12
Step 3: Add the info Callable Object to the process .....	13
1.    Add the Callable object as an infocallable object in the Edit Quotation Step .....	13
2.    At the process level choose the appropriate runtime view.....	14
Running the changed Process .....	16
Variance 2: Adding the Lean Order UI in the first step.....	18
Adding R/3 transaction UI to the composite .....	20
Point to note:.....	23
Related Content.....	24
Copyright.....	25

## Example Scenario – Quotation Approval

To explain the step by step procedure of how to incorporate an ABAP WebDynpro UI, I have taken the Quotation Approval Scenario as example.

The quotation approval composite is an application to aid the work of sales representative who has to prepare a quotation for a customer. The composite guides the sales representative to select a Quotation from the backend ERP system and then create the Quotation letter, edit it if necessary and send it to his manager for approval with a note (or comments).

To simplify the explanation lets look only at the sales representative role and here is the explanation of the steps he would perform. For additional information on this specific scenario see [related content](#).

### Actions Performed by the Sales Representative

#### 1. Select the Quotation from the ERP system.

As shown in the screen shot below the Sales representative initiates the guided procedure process and in the first step selects the appropriate quotation from the ERP system. This UI is created with Visual composer and it uses RFCs to access the data from the ERP system.

Quotation No.	Customer No.	Customer Name	Net Value
0020000019	0000001000	Becker Berlin	53,580.00
0020000020	0000001000	Customer Number lin	44,990.00
0020000021	0000001000	Becker Berlin	859.00
0020000022	0000001000	Becker Berlin	3,868.40
0020000023	0000001000	Becker Berlin	8,091,512.0
0020000024	0000001000	Becker Berlin	11,238.00

#### 2. Add Comments to the Approver.

After selecting the Quotation he can add some notes for his approver.



### 3. Edit the Quotation Letter.

Next he will see an editable document where he can change the text in the letter. The screenshots below show the UI

- ✔ Select/View Quotation
- ✔ CommentsToApprover
- Edit Quotation
- Preview Quotation

ReaderRights credential

Do not show this message again

Pages  
Signatures  
Attachments  
Comments

**Becker Berlin**  
Calvinstrasse 36  
Berlin  
DE

Regarding: Quotation Number -0020000019

Dear Mr/Mrs. Schulz

This is in concern with the proposal discussed with you over the offer you the following prices. I hope you will like....

Phone Number 030-8853-0

Contact Name

**Conditions**

Terms of Payment: \_\_\_\_\_ INCO Term \_\_\_\_\_

Terms of Delivery \_\_\_\_\_

\* Cost Prices and Net Price are in EUR

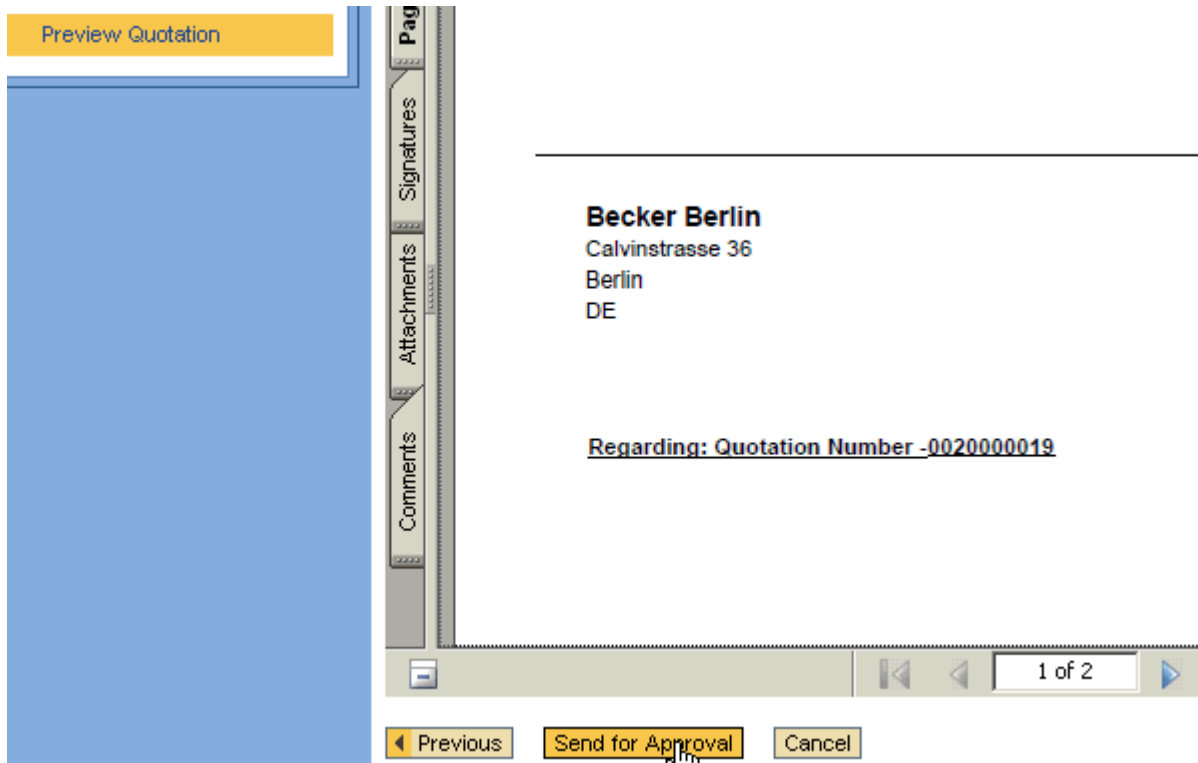
Item No	Material No	Material Description	Unit	Quantity	Cost per Unit
000010	M-06	Flatscreen MS 1460 P	ST	10.00	859.00
000020	M-05	Flatscreen LE 50 P	ST	100.00	449.90

Volum

2 of 2

4. Preview the Created non editable Quotation letter.

Before the Sales representative sends the Quotation to his manager for approval he will be able to see the final created Quotation.



### Enhancement Required

The steps done by the sales representative is straight forward, but at the step where he is editing the Quotation letter (step 3), he wants to change quotation in the backend system. He can launch the lean order ABAP WebDynpro application in a separate window logging into the ERP system and change the Quotation. But instead we can provide him the access to the lean order UI directly from step 3. Let's see how we can do it.

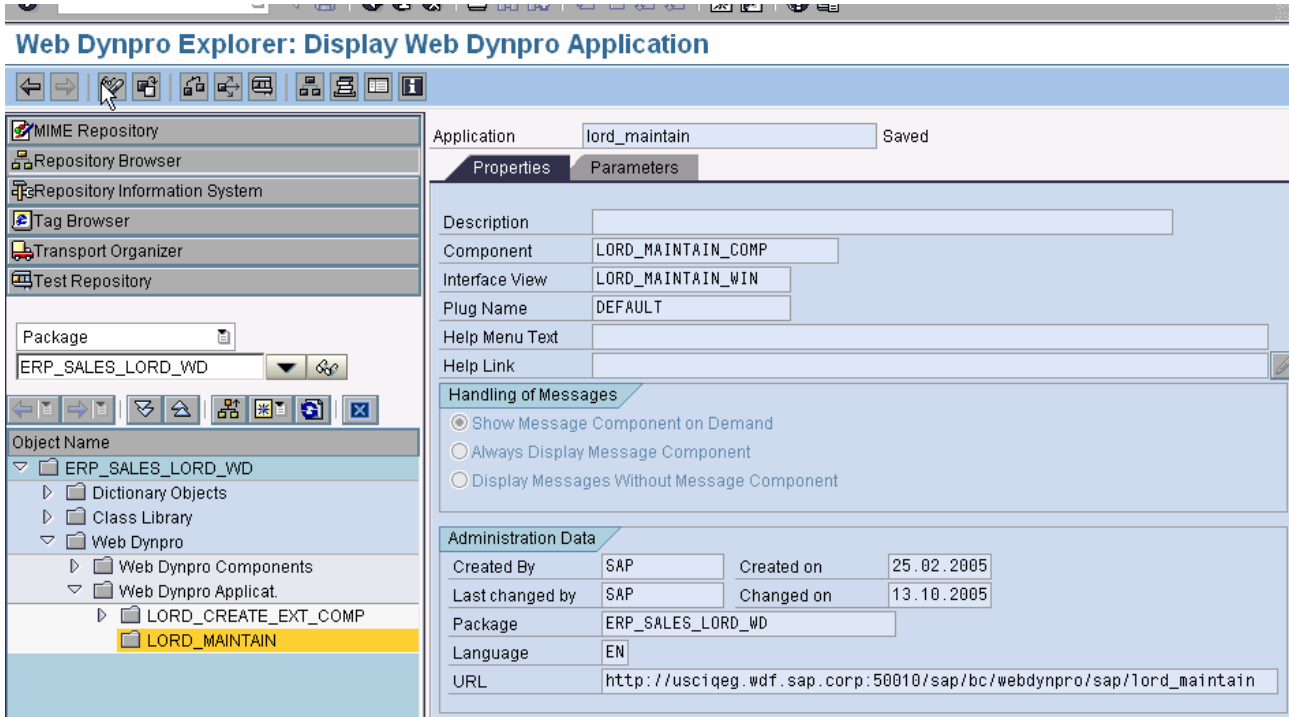
### Adding the Lean Order ABAP WebDynpro UI in the Edit Quotation Step

To make this UI accessible from the composite application, a good way it to add it as an infocallable object in the Guided Procedure step *Edit Quotation*. This infocallable object should of type portal iView. But we should make sure that we pass the Quotation number as a parameter into this infocallable object so that the *Lean Order application* UI opens the Quotation that has been selected by the sales representative. Here are the steps.

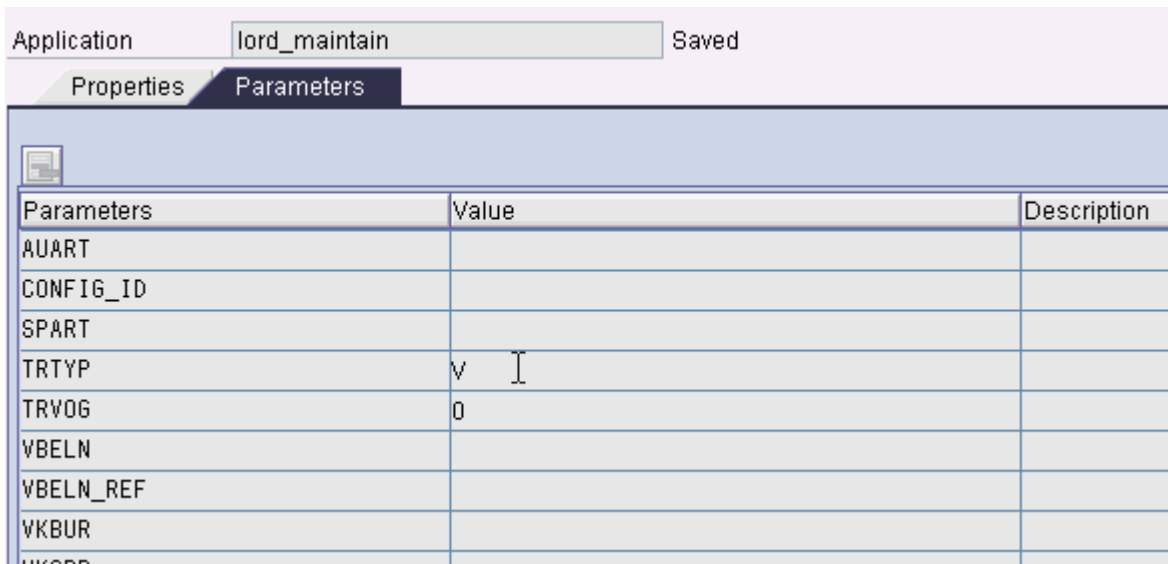
#### Step 1: Create iView

1. Identify the ABAP WebDynpro application.

Identify the ABAP WebDynpro application. Let us open the *Lean Order WebDynpro application* through SE80 in R/3:



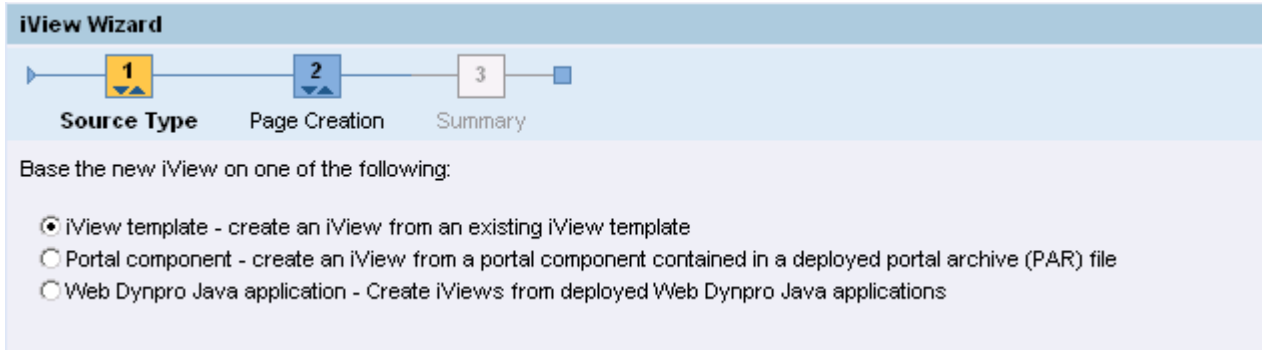
Note the application name – LORD\_MAINTAIN. Open the parameters tab and identify the parameters you need. In our example we need to use the appropriate parameters - to open the application in change mode and with the correct sales document type (QT- quotation). TRYP = V and AUART = QT



## 2. Create a SAP WebDynpro iView in Portal Content Studio.

Next to create the iView goto the portal content studio. The screenshots below explain the procedure:

- Choose iView template.



- Template selection choose – “SAP Web Dynpro iView”

**Step 1: Template Selection**

- KM XML Forms Builder Check iView
- KM XML Forms Builder iView
- Portal Activity Report iView
- Proxy-to-Portlet iView (WSRP)
- Proxy-to-Remote iView
- Related Services iView
- SAP BSP iView
- SAP IAC iView
- SAP MiniApp iView
- SAP RFC iView
- SAP Transaction iView
- SAP Web Dynpro iView
- Service Map iView
- URL iView
- Web Service iView

- Choose “Web Dynpro for ABAP”

**Step 3: Web Dynpro Platform**

Select whether you want to create a Web Dynpro for Java or Web Dynpro for ABAP iView

- Platform: \*
- Web Dynpro for ABAP
  - Web Dynpro for Java

- At the Web Dynpro Parameters – Choose the system (should be defined in the portal system configuration) , Namespace – sap, Application Name – LORD\_MAINTAIN, Application Parameters - TRTYP =V & AUART= QT. These entries will open the Web Dypro application in change mode and with the correct sales document type.



**Step 4: Web Dynpro Parameters**

Enter the parameters of the Web Dynpro application for which you want to create the iView

SAP\_ECC\_Cross ▼

Namespace: \*

Application Name: \*

Web Dynpro Client:  
 Plain HTML  
 Windows Client  
 Java Client

Configuration Name:

Application Parameters:

- After this click “finish” and preview the iView. The iView will open as shown below.

Address  [http://prtroot/pcd!3aportal\\_content!2fCFT\\_Development!2f5DN!2fIviews!2fChangeQuotation?sap-config-mode=true](http://prtroot/pcd!3aportal_content!2fCFT_Development!2f5DN!2fIviews!2fChangeQuotation?sap-config-mode=true)

## Sales Document Processing

 Enter the document number  1 Message 

Start

Transaction type:  ▼

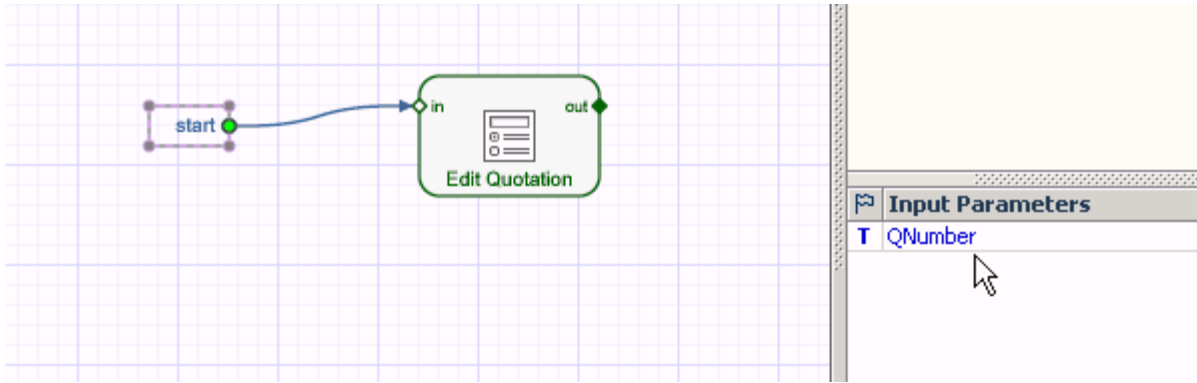
Sales Document:  

- Note down the URL from *pcd!3aportal...* as we will need it later.

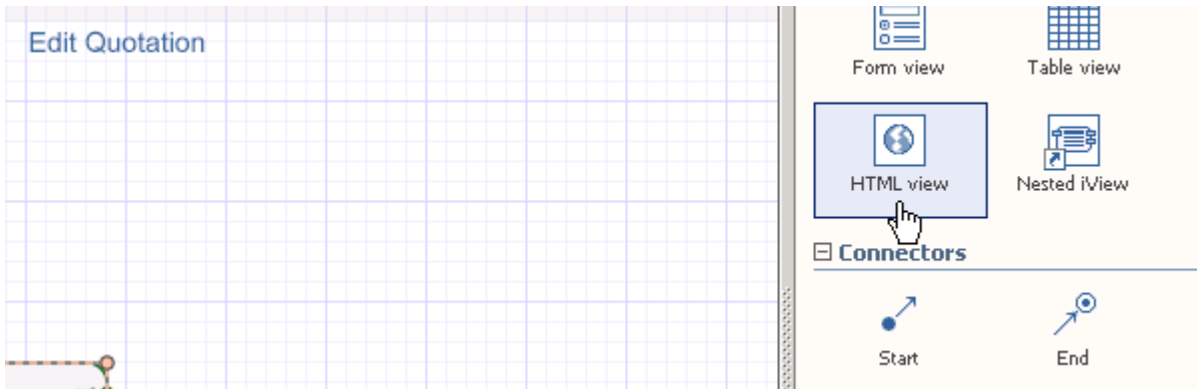
**3. Create the Visual Composer iView.**

Now we have a SAP Web Dynpro iView but the problem is that if we include this iView directly as a GP callable object we will not be able to pass the Quotation Number parameter. To get a GP interface, we will need to “wrap” this iView in a visual composer iView. Let’s see how to do it:

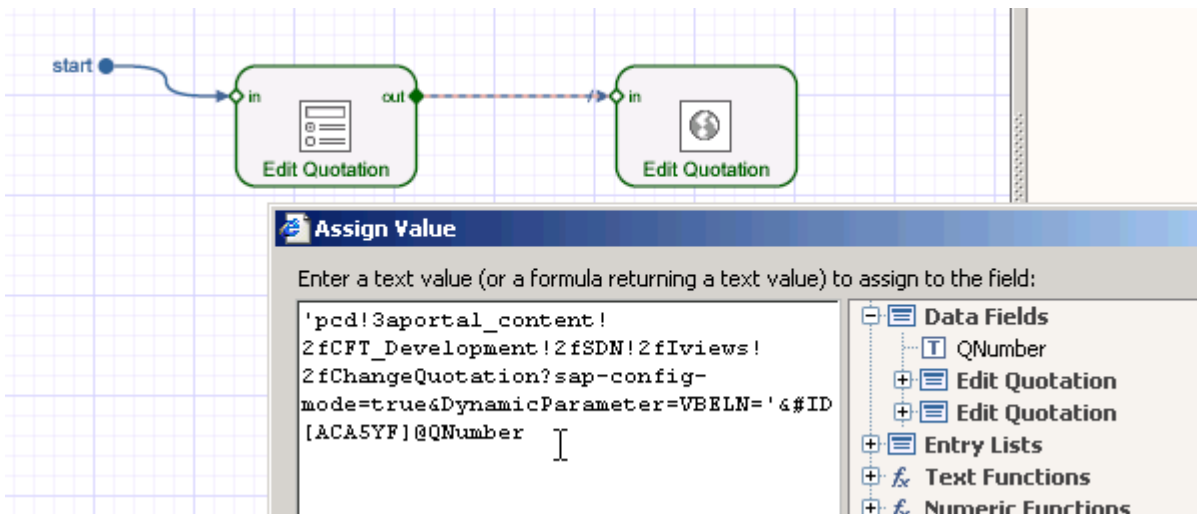
- Open the Visual Composer storyboard and create a new iView. In the iView create a start point with a parameter Qnumber. Drag and drop a form “Edit Quotation” from this start point.



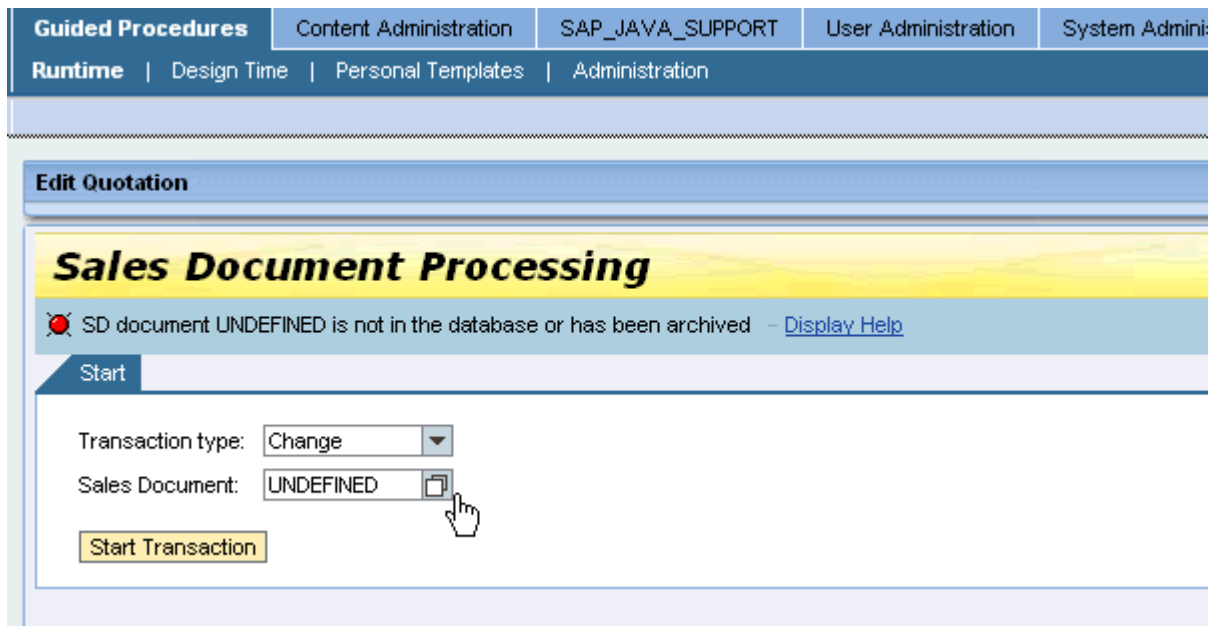
- Now drag and drop into the storyboard a HTML view element and name it as “Edit Quotation”



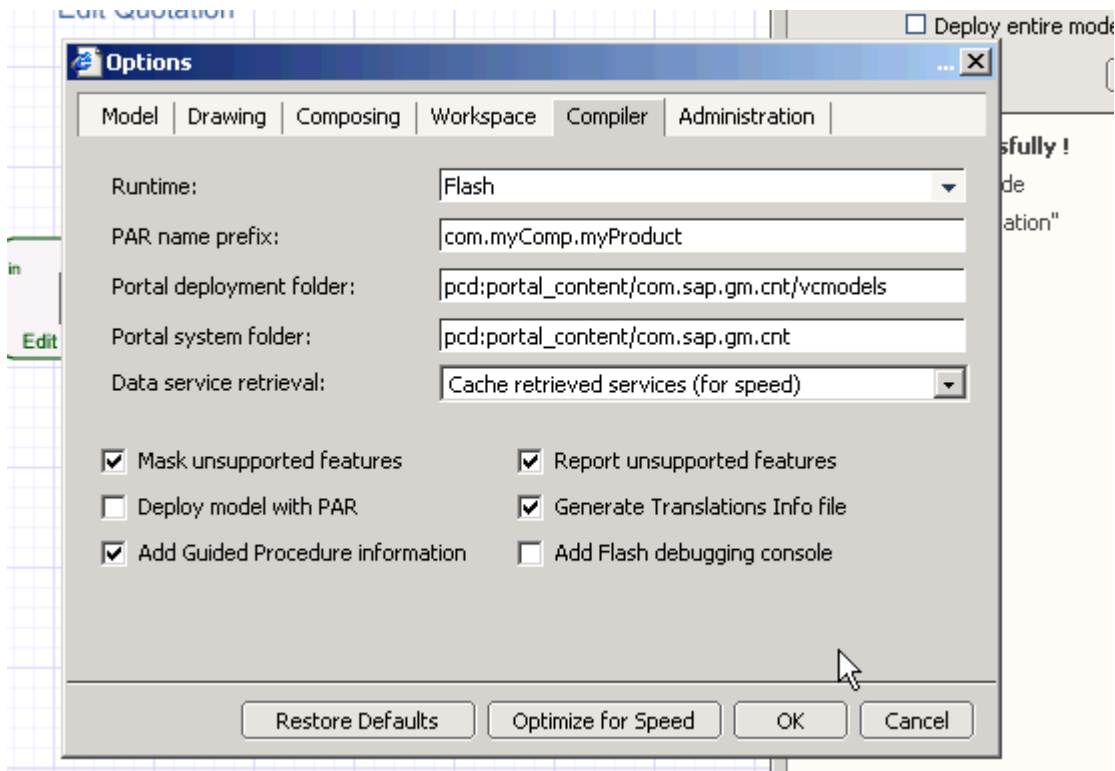
- Connect the out port of “Edit Quotation” form and in port of the HTML view “Edit Quotation” . The HTML view “Edit Quotation” will have an url input parameter by default. Fill in the *url* parameter with the previously noted url from the previous step where we had previewed the iView. Append the url with &DynamicParameter=VBELN= “Qnumber”. This is shown in the screenshot below. This will open the Web Dynpro iView with the Quotation number passed in as a parameter.



- Now deploy the VC iView and run the iView. It will open as shown below. As you can see that “UNDEFINED” is being passed into the Sales Document field. When we include this iView in the GP process and pass the runtime value of the Quotation number then the correct Quotation number will be used.



Note: Before Deploying the VC iView check to see if your Visual composer compiler options are as shown below:



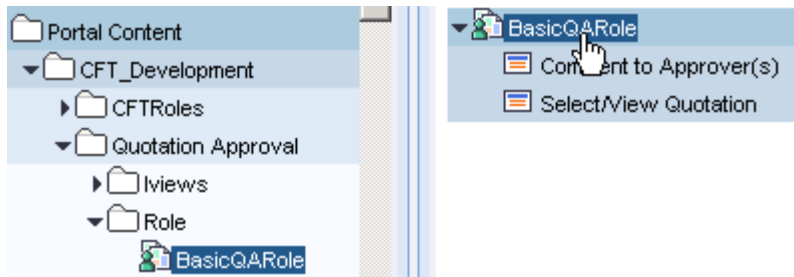
## Step 2: Create the Callable Object

### 4. Add the Visual Composer iView to the Portal role.

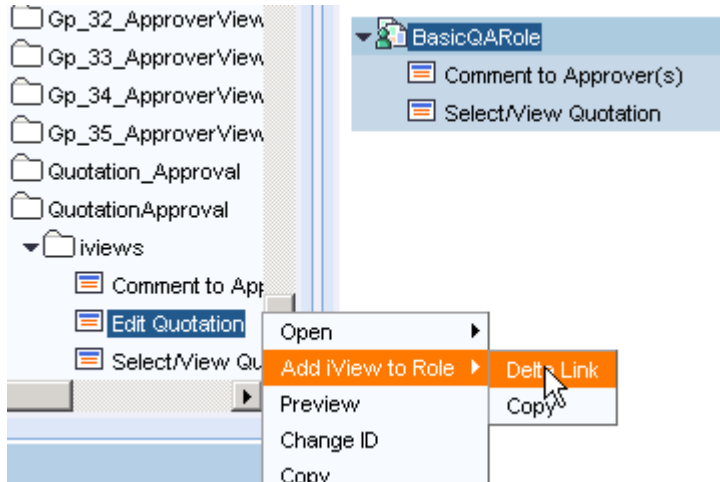
In order to create a portal iView callable object we need to first add our newly created iView, as a delta link, to the role.

The screenshot shows how to do it:

Open the Role:

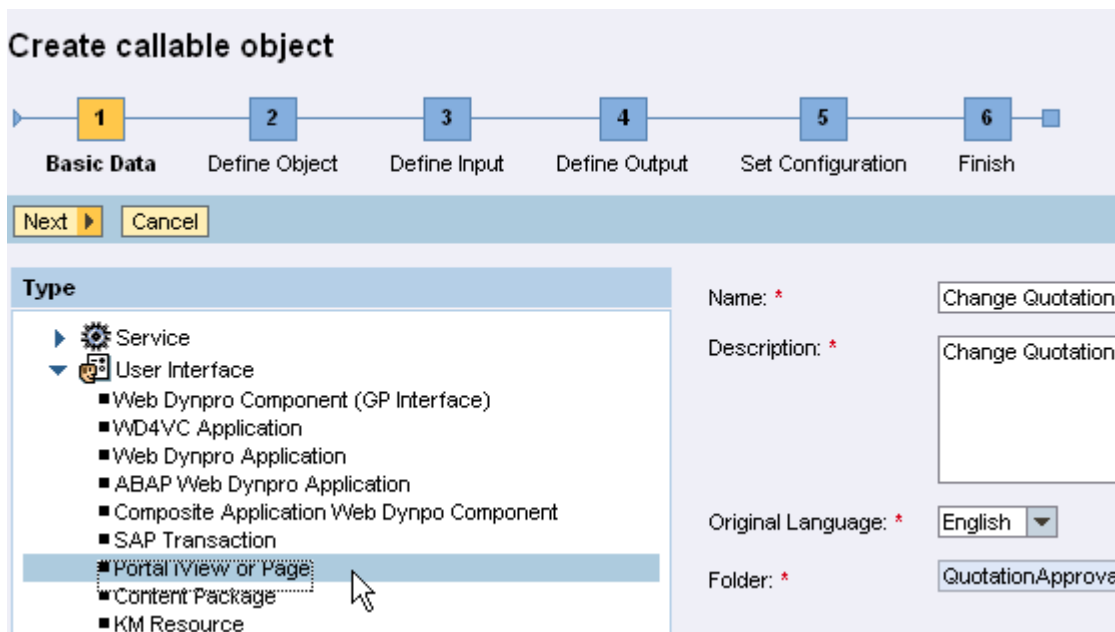


Next add the iView to the role:

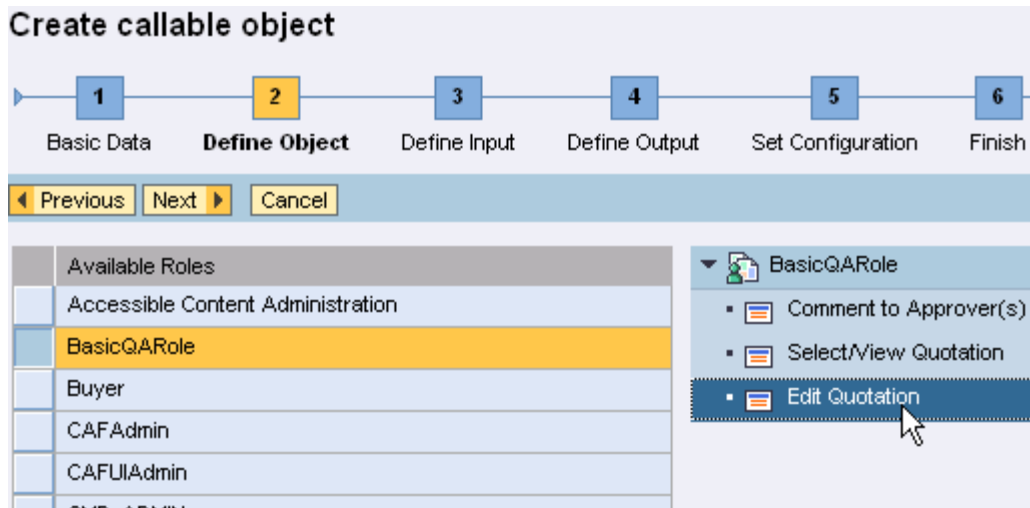


### 5. Create a new Callable Object (CO).

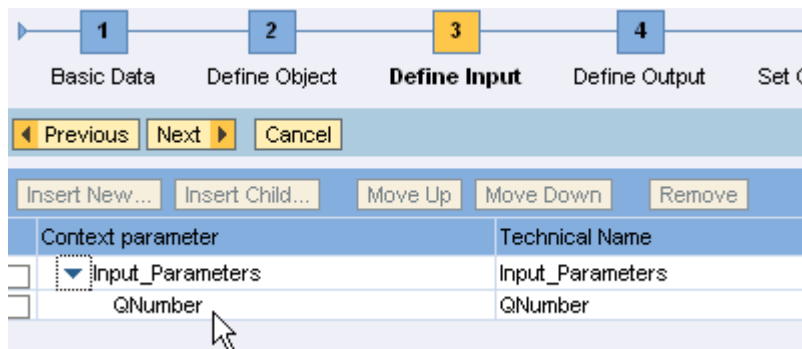
Next we create the callable object - *Change Quotation* of type portal iView. The steps are shown below:



Choose the iView from the Role - *BasicQARole*



In the next step check to see that the input Parameter *Qnumber* is present. If you do not see this input parameter then the Visual Composer iView has not been deployed properly with the correct compiler options. (see Note in Step 1.3)



Click through the next screens and finally activate the callable object. In the “set Configuration” step do not add any result states.

### Step 3: Add the info Callable Object to the process

#### 1. Add the Callable object as an infocallable object in the Edit Quotation Step

Open the process and in the edit quotation step(Action), go to the info callable object tab and add the newly created callable object.

The steps are as shown below.

Name	Type	Target	Item
BasicQuotationApproval	Process		BasicQuotationApp
Block - BasicQuotationAppro...	Sequential Block		Block - BasicQuota
Select/View Quotation	Action		Select/View Quota
CommentsToApprover	Action		CommentsToAppro
GetQuotationDetails	Action		GetQuotationDetails
GetCustomerDetails	Action		GetCustomerDetails
<b>Edit Quotation</b>	<b>Action</b>		<b>Edit Quotation</b>
Result States			
Quotation Letter	Callable Object for Execution		Quotation Letter
Preview Quotation	Action		Preview Quotation

Row 1 of 10

**n: Edit Quotation Edit**

Basic Data | Parameters | Due Date | Permissions | Ad Hoc Items | Attachments | Info Callable Obj

Map Parameters | Set Category | Remove

Callable Object	Type
Change Quotation	com.sap.gp.prt

Next map the parameters, so that the Quotation number value flows into the info callable object from the GP context.

Edit Quotation					Edit Quota
Remove Mapping   Set Default					Parame
Parameter	Object Type	Mapped To	Default	List	QNu
Input_Parameters	Structure			✓	QNe
QNumber	String	QNumber			QNe

2. At the process level choose the appropriate runtime view

Since we have added an info callable object at the Edit Quotation action we will have to choose the appropriate runtime view at the process so that this is seen during the runtime.

The steps are as shown the screenshots below:

Item	Type	Target	Item	Status	Version
BasicQuotationApproval	Process		BasicQuotationApproval		0.5
Block - BasicQuotationApproval	Sequential Block		Block - BasicQuotationApproval		0.4
Select/View Quotation	Action		Select/View Quotation		0.1
CommentsToApprover	Action		CommentsToApprover		0.1
GetQuotationDetails	Action		GetQuotationDetails		0.1
GetCustomerDetails	Action		GetCustomerDetails		0.1
Edit Quotation	Action		Edit Quotation		0.5
Quotation Letter	Callable Object for Execution		Quotation Letter		0.1
Preview Quotation	Action		Preview Quotation		0.2

**BasicQuotationApproval Edit**

Control Items | View Permissions | Attachments | Info Callable Objects | Interactive Forms | **Runtime Views**

Created By:

Choose the runtime view as shown below:

Process Activities	Process Activities (allows to view and execute activities and access activity related information)	sap.com/caf~eu~g
Process Information	Process Information (provides information about the process flow, deliverables and attachments)	sap.com/caf~eu~g
Additional Resources	Additional Resources (provides files, URLs and contacts)	sap.com/caf~eu~g
<b>Process Activities</b>	<b>Process Activities (allows to view and execute activities and access activity related information and start ad-hoc items)</b>	<b>sap.com/caf~eu~g</b>
Overview	Overview (allows to set process options)	sap.com/caf~eu~g
Process Activities	Process Activities (allows to view and execute activities and access activity related information and allows to set process options)	sap.com/caf~eu~g

Make that the default and remove the old view

**Selected Runtime Views**

Add Generic View | Add Custom View | Remove View

Default	View	Description
<input type="radio"/>	Process Activities	Process Activities (allows to view and execute activities)
<input checked="" type="radio"/>	Process Activities	Process Activities (allows to view and execute activities and access activity related information and start ad-hoc items)

With this we are ready to run the process.

## Running the changed Process

On re running the process the Edit Quotation step looks like as shown below. The info callable object Change Quotation is available under the options category.

The screenshot displays the SAP Web Dynpro interface. On the left, the 'Process Activities' sidebar is visible, containing three main sections: 'Activities', 'Options', and 'You Can'. Under 'Activities', there are four items: 'Select/View Quotation' (checked), 'CommentsToApprover' (checked), 'Edit Quotation' (highlighted in orange), and 'Preview Quotation'. The 'Options' section contains a single item, 'Change Quotation', which is highlighted in blue and has a mouse cursor pointing to it. The 'You Can' section states 'There are no additional activities that you can carry out'. The main content area on the right shows a document preview with a toolbar at the top (including zoom, pan, and search icons) and a 'Document Rights and Instructions' section. Below this, the text 'ReaderRights credential' is displayed, followed by a checkbox for 'Do not show this message again'. The document content itself is a letter from 'Becker Berlin' at 'Calvinstrasse 36, Berlin, DE', with the subject line 'Regarding: Quotation Number -0020000019'.

On clicking on the change quotation info callable object the WebDynpro will open the appropriate quotation in “change mode” (shown below). On clicking *back to view* the Quotation letter will be shown. In this particular example the Edit Quotation letter is a WebDynpro that uses the adaptive RFCs to fetch the quotation details, so when one goes back to the view the changed quotation details will be fetched again.



**Process Activities**

- Activities
  - Select/View Quotation
  - CommentsToApprover
  - Edit Quotation**
  - Preview Quotation
- Options
  - Change Quotation
- You Can
 

There are no additional activities that you can carry out

**Edit Quotation**

**Change Quotation 20000019**

Sales area: Germany Frankfurt, Final customer sales, Cross-division

Buttons: Save Document, Delete Document, Display, Submit

**Document Main View**

- Header Details
- Incompletion Log
- Item Details
- Related documents
  - Preceding Documents: Sales Documents
  - Subsequent Documents: Sales Document, Deliveries

**Header**

Quotation: 20000019  
 Sold-to party: \* 1000  
 Ship-to party: 1000

Buttons: Show Header Details

**Items**

Buttons: Add new Lines, Delete Item

Detail	Item	M
<input type="checkbox"/>	10	M

In this manner we were able to integrate an already existing WebDynpro application into a composite application and more importantly pass the context values.

## Variance 2: Adding the Lean Order UI in the first step.

If the requirement is to add the lean order WebDynpro UI to the first step where sales representative chooses the Quotation and wants to edit it directly there. Then following a very similar approach this can be easily achieved without any modifications to the GP Process, but only to the VC model of the first step. We could use another layer in the VC model to have the Lean Order iView included as a HTML View. I am not going to describe the steps in detail here but it can be easily achieved with the know-how gained in the previous exercise. The picture of the new UI is as shown below:

The screenshot displays a SAP Web Dynpro interface. On the left is a sidebar with a list of activities under the heading 'activities'. The first activity, 'Select/View Quotation', is highlighted in orange. Below this are several other activities: 'Comment to Approver(s)', 'Edit Quotation Letter', 'Preview Quotation Letter', 'Rework Quotation', and 'Edit Quotation'. Underneath the activities list is a section titled 'Options' containing a button labeled 'Create Quotation'. Below that is a section titled 'You Can' with a message: 'There are no additional activities that you can carry out'. The main content area is titled 'Selection Criteria' and contains several input fields: 'Customer:', 'Created By:', 'Valid From:', 'Valid To:', and 'Status:'. The 'Status' dropdown is currently set to 'Not approved' and has a red asterisk next to it. Below the selection criteria is a table titled 'Selected Quotations' with the following data:

Quotation No.	Customer No.	Customer Name	Net Value
0020000019	0000001000	Becker Berlin	535
0020000020	0000001000	Becker Berlin	449
0020000021	0000001000	Becker Berlin	8
0020000022	0000001000	Becker Berlin	386
0020000023	0000001000	Becker Berlin	80915
0020000024	0000001000	Becker Berlin	112

At the bottom of the main content area, there are three buttons: 'Edit Quotation', 'View Quotation', and 'Comment To Approver'. A mouse cursor is pointing at the 'Edit Quotation' button.

**Activities**

- Select/View Quotation
- Comment to Approver(s)
- Edit Quotation Letter
- Preview Quotation Letter
- Rework Quotation
- Edit Quotation

---

**Options**

- Create Quotation

---

**Can**

no additional activities that carry out

**Edit Quotation**

Back

Change Quotation 20000019

**Sales area** Germany Frankfurt, Final customer sales, Cross-division **Net**

**Document Main View**

- Header Details**
- Incompletion Log**
- Item Details**

▼ **Related documents**

**Preceding Documents**

- Sales Documents

**Subsequent Documents**

- Sales Document
- Deliveries
- Billing Documents

▼ **Related Services**

**Create new Document**

- Inquiry ▶
- Quotation ▶
- Order ▶

**Create subsequent document**

**Header**

Quotation:

Sold-to party: \*

Ship-to party:

Show Header Details

**Items**

	Detail	Item	Material
<input type="checkbox"/>		10	M-06
<input type="checkbox"/>		20	M-05

## Adding R/3 transaction UI to the composite

Similar to the method described for access the SAP ABAP WebDynpro applications SAP R/3 transactions also can be accessed. The iView that needs to be created initially is of a different type. The steps to create this iView are as shown below:

Here I have taken the example of Change Quotation transaction (VA22).

Create a new iView in the portal content studio:

- Choose iView template.

Base the new iView on one of the following:

- iView template - create an iView from an existing iView template
- Portal component - create an iView from a portal component contained in a deployed portal archive (PAR) file
- Web Dynpro Java application - Create iViews from deployed Web Dynpro Java applications

- Choose the iView template type as “SAP Transaction iView”

### iView Wizard

#### Step 1:

#### Template Selection

Choose a template:

- KM XML Forms Builder Check iView
- KM XML Forms Builder iView
- Portal Activity Report iView
- Proxy-to-Portlet iView (WSRP)
- Proxy-to-Remote iView
- Related Services iView
- SAP BSP iView
- SAP IAC iView
- SAP MiniApp iView
- SAP RFC iView
- SAP Transaction iView
- SAP Web Dynpro iView
- Service Map iView
- URL iView
- Web Service iView
- Workset Map iView
- XML iView

- Name the iView appropriately:

**iView Wizard**

**Step 2: General Properties**

iView Name: \*

iView ID: \*

iView ID Prefix (Example: com.companyname):

Master Language: \*

Description:

- In the next screen choose the option “SAP GUI for Windows”

**iView Wizard**

**Step 3: SAP GUI Type**

Enter the SAP GUI type for which you want to create the iView

SAP GUI Type: \*  
 SAP GUI for HTML  
 SAP GUI for Windows  
 SAP GUI for Java

- Choose the portal system from the list and also enter the SAP transaction.

**iView Wizard**

**Step 4: Transaction Parameters**

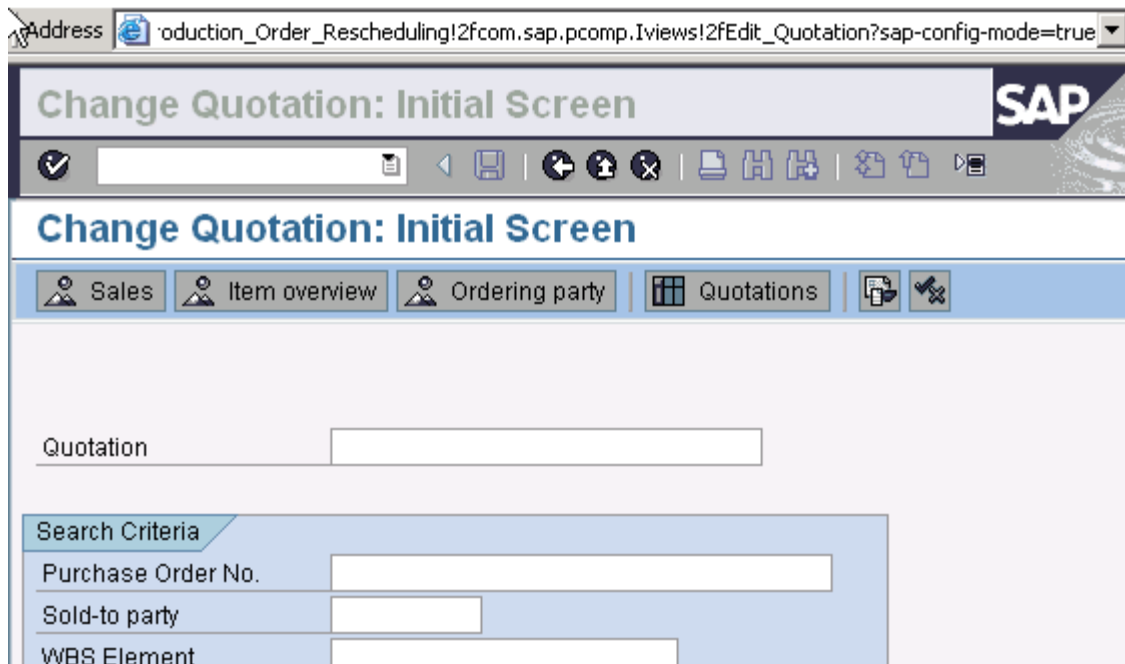
Enter the parameters of the SAP transaction for which you want to create the iView

System: \*

Transaction Code: \*

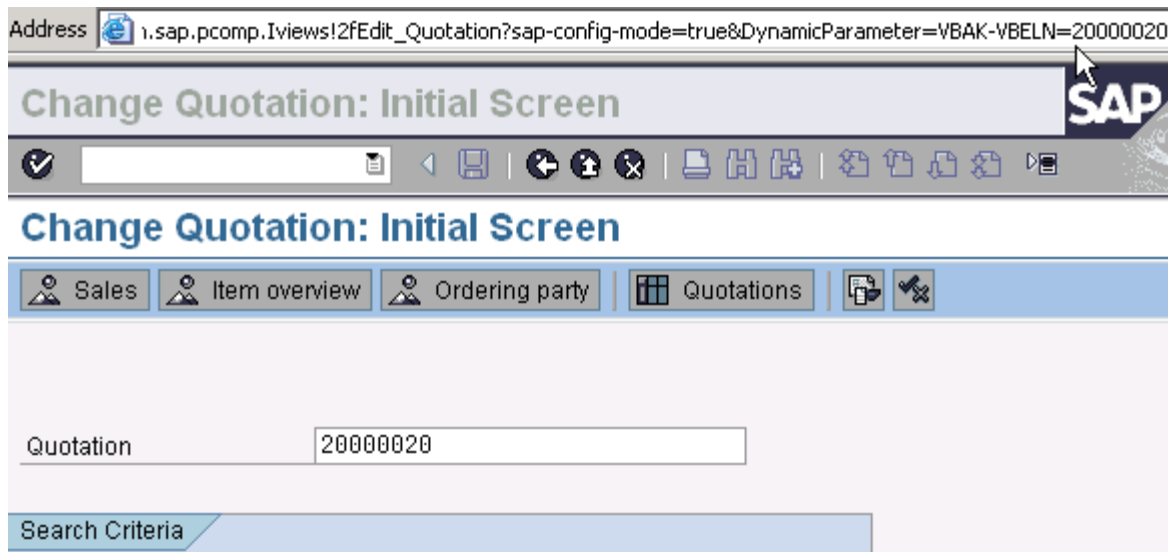
Application Parameters:

- Click through the next screens and then preview the iView. You will see a screen similar as below.



Now you have the iView ready and you can follow the steps ([step1. 3 Create the Visual Composer iView..](#)) as mentioned above to continue to create the VC iView and incorporate it later in the process.

To pass the parameters into this R/3 transaction from the GP context follow a similar approach by appending the Dynamic parameter part to the URL, as shown in the screen below use the parameter VBAK-VBELN to pass the Quotation number.



Note: To find the correct parameter names, go to the R/3 transaction and check the technical information of the field you need to populate:

Quotation	
<b>Technical Information</b>	
<b>Search Criteria</b>	
Purchase Order No.	
Sold-to party	
WBS Element	
<b>H</b> Search	
<b>Screen Data</b>	
Program Name	SAPMV45A
Screen number	0102
<b>GUI Data</b>	
Program Name	SAPMV45B
Status	A22
<b>Field Data</b>	
Table Name	VBAK
Field Name	VBELN
Search Help	VMVA
Data Element	VBELN_VA
DE Supplement	0
Parameter ID	AUN
<b>Field Description for Batch Input</b>	
Screen Field	VBAK - VBELN

### Point to note:

The method described here can also be achieved using the SAP transaction callable object type and the ABAP WebDynpro callable object type. Using the callable object types is a more standard way of achieving the same as GP documentation describes it. But there are some subtle differences and let's discuss some pro and cons of these methods.

Using the GP callable object type method is more standard and it automatically selects the interface for passing the parameters into the backend UI, whereas the iView method described in this document is more tedious approach, especially finding out the correct parameters is a manual process. But there are subtle advantages – the most important being that you will not have to define additional end points in GP for the backend system. So if your composite is using iViews then one will have to create the portal end points and GP backend system end points, which is a bit annoying. So in order to totally avoid GP end point definitions one could use this method so that all you have to do is use the portal system alias uniformly through out your composite application and while installing one have to just create the one portal system connection for each backend system (one for ERP one for CRM etc). The other big advantage of this method is integrating the ABAP WebDynpro directly in a VC UI as described in the section [Variance 2: Adding the Lean Order UI in the first step](#). This may not be possible if you were to create a different callable object to access the lean order UI. There are also other minor differences like in the iView method you could choose to use of the following ways to display – SAP GUI for Java, SAP GUI for HTML or SAP GUI for Windows, where in the SAP transaction callable object type one uses one standard type.

Since this article describes one more method to integrate SAP transactions and ABAP WebDynpro one could use any of the methods either using the callable object type or following the procedure described here based on the requirement.

## Related Content

Check out of the box demonstration of SAP xApps for Personal Productivity from the ESA discovery server. The quotation approval composite is explained more in detail here also see other composite examples.

- <http://sdn.sap.com/irj/sdn?rid=/webcontent/uuid/b390ac85-0d01-0010-c186-887f0dea5db2>

Learn how Guided Procedures (GP) helps you to build composite applications

- <http://sdn.sap.com/irj/servlet/prt/portal/prtroot/docs/library/uuid/0b1c2eda-0c01-0010-b597-b107502cdc71>

Learn how you can use SAP R/3 transactions as Callable Objects in Guided Procedures.

- <https://www.sdn.sap.com/irj/servlet/prt/portal/prtroot/docs/library/uuid/c48691d1-0c01-0010-9380-f98819c3ccc7>

Learn how you can use ABAP WebDynpro as Callable Objects in Guided Procedures

- <https://www.sdn.sap.com/irj/servlet/prt/portal/prtroot/docs/library/uuid/9cbabed1-0c01-0010-dba5-bdddbadfe586>



## Copyright

© Copyright 2006 SAP AG. 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. Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, Informix, I5/OS, POWER, POWER5, OpenPower and PowerPC are trademarks or registered trademarks of IBM Corporation.

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

Oracle is a registered trademark of Oracle Corporation.

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

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

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

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. 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.

These materials are provided "as is" without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

Any software coding and/or code lines/strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.