

# Integrating Web Dynpro and SAP NetWeaver Portal

## Part 1: Creating Web Dynpro-Based Portal Content

### Applies to:

Integration of Web Dynpro for Java applications into SAP NetWeaver Portal for the SAP NetWeaver 04s SP8 release.

### Summary

This article describes the creation of Web Dynpro for Java iViews on top of the Web Dynpro page builder. In addition, differences in iViews created within an SAP NetWeaver 04 portal are discussed.

**Author(s):** Jochen Guertler

**Company:** SAP AG

**Created on:** 26 June 2006

### Author Bio



Jochen Guertler works as a development architect within the Web Dynpro for Java team. His main responsibilities are the integration of Web Dynpro for Java with other components of SAP NetWeaver, especially the integration with the SAP NetWeaver Portal.

Jochen is co-author of the book "Maximizing Web Dynpro for Java" from SAP Press.

Part 1: Creating Web Dynpro-Based Portal Content.....	1
Applies to: .....	1
Summary.....	1
Author Bio .....	1
Introduction .....	3
Prerequisites .....	4
Web Dynpro for ABAP .....	4
Implementing Web Dynpro-Based Portal Content .....	4
The Web Dynpro Page Builder .....	5
The Web Dynpro Page.....	6
Creating a Web Dynpro Page .....	6
Using the Page Editor.....	9
Differences between a Web Dynpro Page and a PRT Page .....	9
Creating a Web Dynpro iView.....	10
Relationship to Web Dynpro iViews Created within SAP NetWeaver 04 .....	10
Black Box iViews vs. Split Applications.....	11
Creating a Black Box iView .....	11
Defining Several iViews Based on One Web Dynpro Application .....	14
Exposing a Static List of iViews .....	19
Using the SeveraliViewsApp Example Application .....	19
Defining the iViews .....	20
Launching a Web Dynpro iView .....	22
Further Information .....	24
Copyright.....	25

## Introduction

Although Web Dynpro applications can generally run as standalone applications, there are several benefits to running Web Dynpro applications in SAP NetWeaver Portal.

The main goal of this series is to give a comprehensive update about the available functionality and possibilities to run a Web Dynpro for Java application in SAP NetWeaver Portal on top of the SAP NetWeaver 04s release.

Besides documentation we will provide running examples to demonstrate the described features.

We will discuss the following topics within this series:

### 1. Creating Web Dynpro-based portal content

In the first part of the series we describe how to create Web Dynpro Java portal content based on the *Web Dynpro page builder*. We discuss the benefits of using the Web Dynpro page builder and we explain the differences to the SAP NetWeaver 04 Web Dynpro iViews.

In the end, we give some more details about the role of the SAP application integrator to launch the Web Dynpro Java application.

### 2. Using portal navigation within a Web Dynpro application

This part covers the navigation between different Web Dynpro applications running in the portal. Besides the standard navigation features we discuss also the possibility to suspend a Web Dynpro application when navigating to another iView.

### 3. Using Object-Based Navigation (OBN) within a Web Dynpro application

In this part we discuss the possibilities using the OBN within a Web Dynpro application running in the portal.

### 4. Using portal eventing within a Web Dynpro application

This part explains how to use portal eventing as a simple communication channel between several Web Dynpro applications running on the same portal page.

### 5. Using the portal "work protect mode" within a Web Dynpro application

Within this part we discuss the usage of the portal work protect mode to make sure that no unsaved data is lost when running your Web Dynpro application in the portal.

## 6. Accessing portal services within a Web Dynpro application

Accessing portal services within your Web Dynpro application gives you a lot of new possibilities and features. In this part of the series we discuss in detail also the portal services provided by the Web Dynpro page builder.

## 7. Manipulating the theme used by a Web Dynpro application

To get a consistent user interface, a Web Dynpro application running in the portal uses automatically the defined portal theme. In this part we explain the details.

## 8. Accessing user data within a Web Dynpro application

In this part we describe how to access user data like the assigned roles within your Web Dynpro application.

## 9. Personalizing a Web Dynpro application

One of the most exciting new features within the NetWeaver 04s release is the personalization of a Web Dynpro application running in the portal. In this part we describe the differences between the explicit and implicit personalization and drill down the given possibilities.

## 10. Using extended features of the SAP Application Integrator

As with NetWeaver 04 the SAP Application Integrator is also within NetWeaver 04s responsible to launch a Web Dynpro application when running in the portal. In this part we discuss the steps, which are executed by the SAP Application Integrator. Furthermore we describe some extended features of the SAP Application Integrator.

## 11. Running Web Dynpro based portal content within your system landscape

In the last part of this series we give an overview about possible and recommended scenarios how to run your Web Dynpro Java based portal content within your system landscape. We explain how to separate the portal and your Web Dynpro applications in different installations and we discuss the details how to use the Federated Portal Network (FPN) for that.

### Prerequisites

Although most of the described functionality is available with the SAP NetWeaver 04s SP5 shipment, the whole series is based on the SAP NetWeaver 04s SP8 shipment. All provided examples and test projects are tested with this release.

In some areas there are minor extensions planned for further releases. We will mention this explicitly.

### Web Dynpro for ABAP

For SAP NetWeaver 2004s, the Web Dynpro based page builder and all depending functionality is only available for Web Dynpro for Java applications. Web Dynpro for ABAP applications still use the URL-based integration available since SAP NetWeaver 2004. Therefore all described functionality in this series is only valid for Web Dynpro for Java. From here on, "Web Dynpro" means "Web Dynpro for Java."

### Implementing Web Dynpro-Based Portal Content

Implementing content for the SAP NetWeaver Portal using Web Dynpro creates different requirements on at least two different layers:

On one side there is the need to build real applications on a real "application layer" – to do that, you need things like:

- Tight-coupled components running in the same session / transaction scope
- Powerful mechanisms for data sharing and communication between these components
- Extended navigation mechanism inside one application (including advanced possibilities to pass parameters from the navigation source to the navigation target).

On the other side all these applications are put together in the SAP NetWeaver Portal on the “portal layer”. Doing that you need for example:

- Loosely-coupling of content based on different UI technologies
- Grouping and structuring content based on different UI technologies in a consistent and generic way
- Role-based customization and / or end-user personalization of the complete content (both regarding application internal settings and more UI related aspects like the layout of grouped content).

Although Web Dynpro applications are in SAP NetWeaver 2004 well integrated into the SAP NetWeaver Portal the boundaries between the “application layer “Web Dynpro and the “portal layer” were clearly visible for the application developer, the portal content administrator and the end user.

As one result of this separation there was often the need to decide in which layer you want to implement your stuff. And of course the result was in most cases “sub-optimal” solution like the PRT-based MSS scenario where the “loosely coupled” portal layer was used to implement a “tight-coupled” real application. Other examples are the Guided Procedures (GP) and Composite Application Framework (CAF) frameworks, where on the Web Dynpro “application layer” a lot of features of the “portal layer” (like generic features for navigation and structuring of content) were re-implemented, as these frameworks are heavily depending on the tight-coupled “application layer” but want to provide some typical “portal layer” features also.

At least with the new people-centric UI design like Control Centers and Work Centers both layers merge together more and more and the complete solution depends on a tight integration of both layers.

The following list summarizes the most obvious restrictions based on the separation between the “application layer” and the “portal layer” in SAP NetWeaver 04:

- Web Dynpro iViews could be only embedded as isolated iViews. Putting more than one Web Dynpro iView on the same page could cause performance problems.
- The portal does not know anything about the “internal” structure of a Web Dynpro application (running in one Web Dynpro iView).
- During the creation of Web Dynpro based portal content there is no knowledge about the available (i.e. deployed) Web Dynpro content. Therefore there are no useful value helps available.
- Communication between a Web Dynpro iView and the portal is done via client-side mechanisms, such as portal-eventing. Even in cases whereby both the Web Dynpro application and the portal itself is running on the same J2EE engine there is no server-side communication channel available.

## The Web Dynpro Page Builder

To overcome the above mentioned restrictions and boundaries of the integration based on SAP NetWeaver 04, one main step within the SAP NetWeaver 04s release is the introduction of the *Web Dynpro page builder*.

From a technical point of view the Web Dynpro page builder is a standard Web Dynpro application. In general the Web Dynpro page builder handles two main aspects:

- Launch a Web Dynpro application running as a *Web Dynpro iView*. The Web Dynpro page builder therefore provides the *portal environment* which defines for example the used theme or other portal

settings used by the Web Dynpro runtime when running an application inside the portal (i.e. as Web Dynpro iView). Using the *portal service factory*, which is part of the provided portal environment, the launched Web Dynpro application has access to a set of portal services (we will describe these services later in this series).

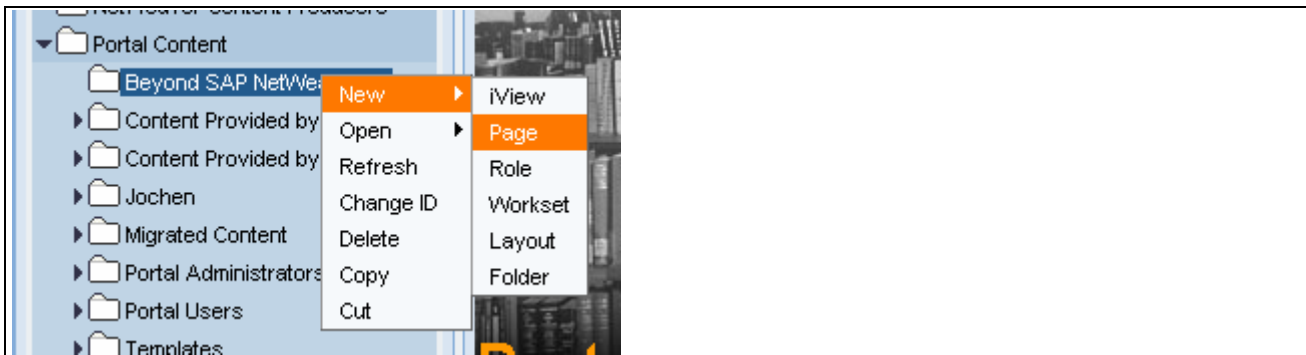
- Create, run and layout a portal page depending on the page definition stored in the Portal Content Directory (PCD). In the following we will name pages executed by the Web Dynpro page builder as *Web Dynpro pages*. In chapter 0 we will describe the benefits and restrictions of a Web Dynpro page in detail. Furthermore we will discuss the relationship to the standard PRT pages executed by the PRT based page builder.

## The Web Dynpro Page

As written in the last chapter, a Web Dynpro page is a portal page executed by the Web Dynpro page builder. From a technical point of view a Web Dynpro page is a standard portal page. You can create a Web Dynpro page using the standard page wizard and you can maintain a Web Dynpro page using the standard page editor. On the other side, there are some differences to standard portal pages executed by the PRT-based page builder (from now on we will refer to these pages as "PRT pages").

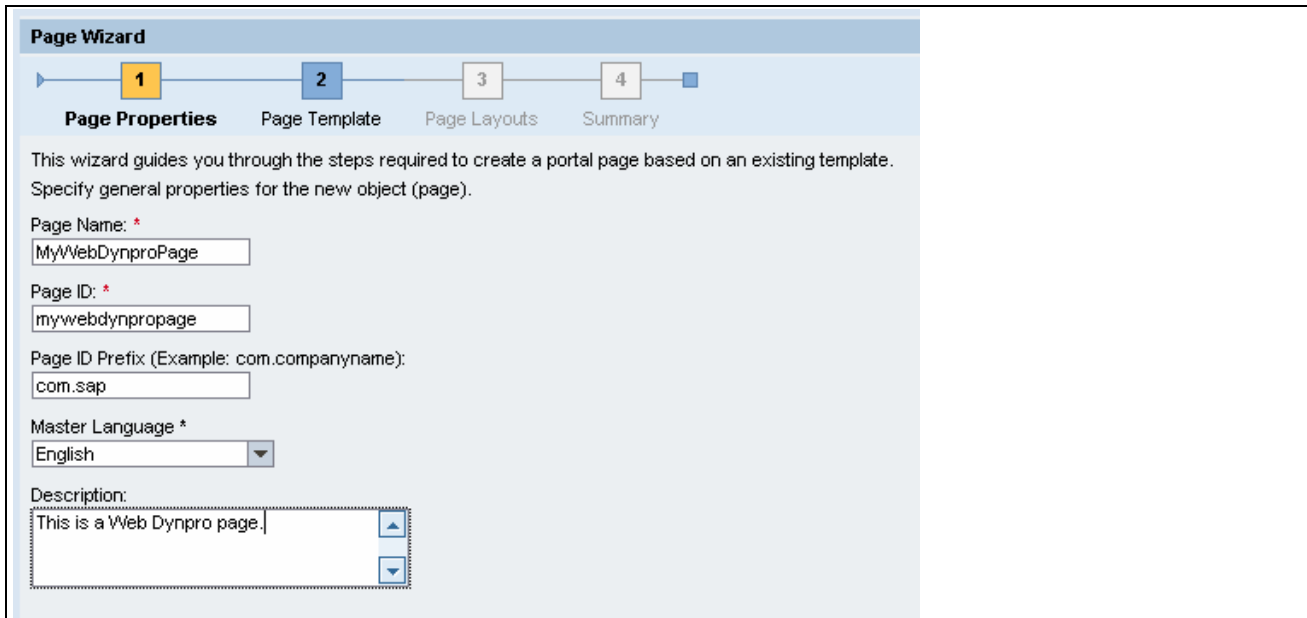
### Creating a Web Dynpro Page

To create a Web Dynpro page, open the Portal Content Studio by navigating to **Content Administration -> Portal Content**. In the Portal Catalog, start the page wizard by choosing **New -> Page** in the context menu of a folder in which you want to create the page.



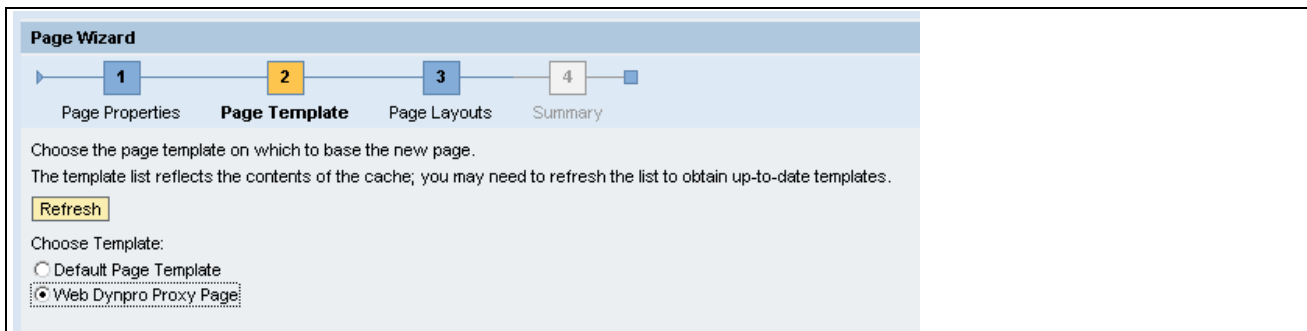
### Starting the Web Dynpro page wizard

On the first screen you have to define some basic page properties like the page name, the page ID or the page ID prefix.



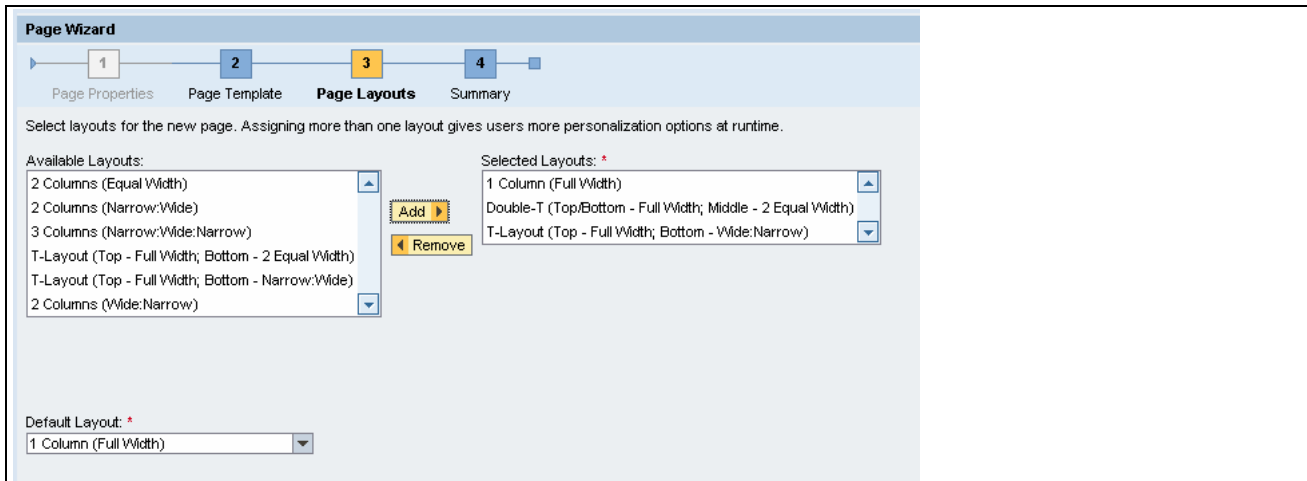
### *Defining the basic page properties*

After pressing the **Next** button you have to choose the needed page template. To create a Web Dynpro page you have to choose the **Web Dynpro Proxy Page** option. Using the **Default Page Template** you would create a standard PRT page.



### *Selecting the Web Dynpro page template*

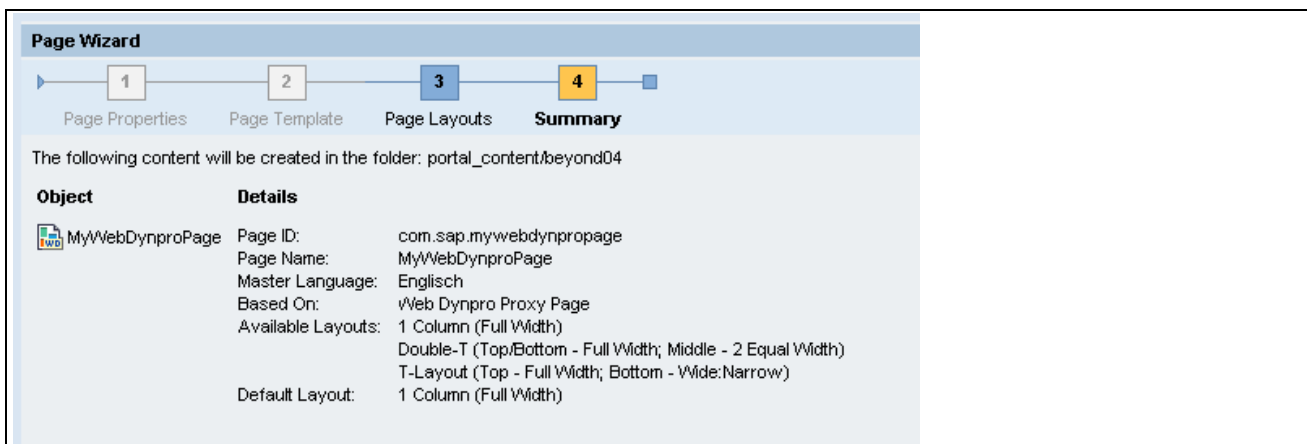
Pressing the **Next** button after that brings you to the next step, where you define the needed page layout(s) for the new page. You can define the same page layout(s) both for a PRT page and a Web Dynpro page.



### Defining the needed page layouts

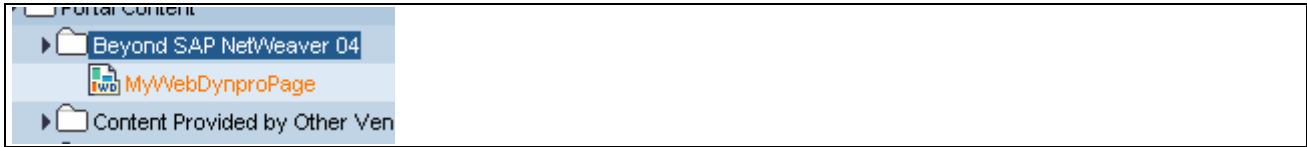
Within the SAP NetWeaver 04s release it is *not* possible to define application specific page layouts for Web Dynpro pages. Within the SAP NetWeaver 07 release this will be possible. Layouts for Web Dynpro page are implemented then as Web Dynpro components implementing a special `ILayout` Web Dynpro component interface.

The last step of the page wizard shows an overview of the created objects. If everything is ok for you the new Web Dynpro page is created after pressing the **Finish** button.



### Summary of created objects

The new `MyWebDynproPage` page is created in our example in the `Beyond SAP NetWeaver 04` folder. To make it visible that this is not a standard PRT page the used page icons contains a small `WD`.



*Visualization of a Web Dynpro page in the content catalog*

### Using the Page Editor

Using the *page editor* you could modify any property of the selected page and the page layout (i.e., the arrangement of iViews running on a page). You start the page editor using the **Open -> Object** context menu of the page object.

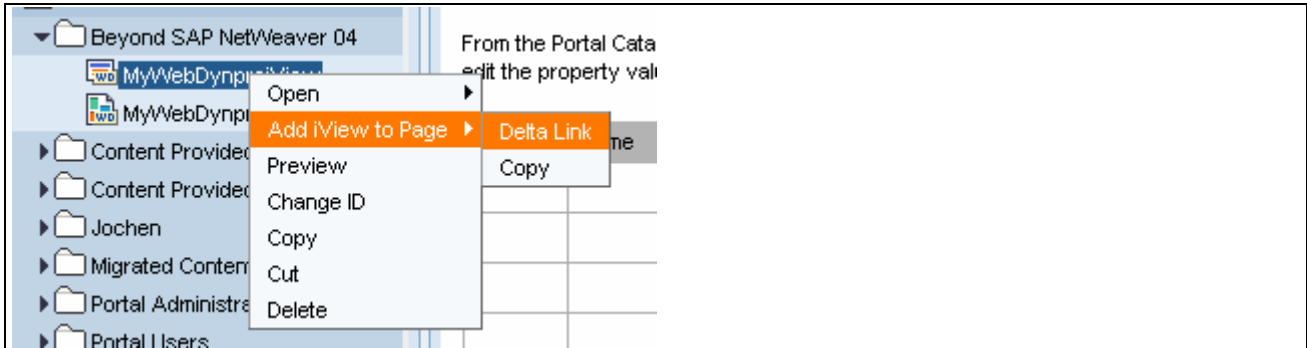


### *Opening the page editor*

The page editor can be used both for standard PRT and Web Dynpro pages.

To add an iView (or another page) to a page you use the **Add iView to Page -> Delta Link** context menu entry of the iView, which should be added to the selected page.

We recommend adding the iView as a delta link to make sure that you do not loose the modifications of the iView itself.



### *Adding an iView to a page*

#### Differences between a Web Dynpro Page and a PRT Page

As long as you want to combine several Web Dynpro iViews on one page, you should always use a Web Dynpro page. By doing this you have the following benefits:

- Web Dynpro iViews run as embedded iViews without any usage of iFrames. Therefore the whole page is refreshed at once without annoying flickering of the single iViews.
- As the Web Dynpro page is processed by the Web Dynpro page builder, all page related UI parts like the iView Trays are rendered as standard Web Dynpro UI. This includes the possibility to run this page in the future in one of the upcoming new Web Dynpro client (like the Windows Smart Client or the Flex-client) without any changes.

But as always – there are also some restrictions using a Web Dynpro page:

- There is no way to run non Web Dynpro iViews as embedded iViews. This is true both for the SAP NetWeaver 04 version of the Web Dynpro iView and also for any other type of iViews. Especially for iViews based on a portal component this could create some problems, as the PRT page supports this.

The general rule quite basic: the Web Dynpro page should be used whenever possible. Only for special scenarios, such as embedding of PRT iViews, should you use a PRT page.

Also the combination of both page types is possible. You can add a PRT page to a Web Dynpro page as isolated page. Adding a Web Dynpro page to a PRT page is also possible. We will discuss possible scenarios and some recommendations in part 11 of this series.

## Creating a Web Dynpro iView

Web Dynpro applications cannot be started automatically within SAP NetWeaver Portal. To make them available, the first step is always to create an associated *Web Dynpro iView*.

By introducing the Web Dynpro page builder we also introduce a new version of Web Dynpro iViews within the SAP NetWeaver 04 release. The main difference between the two releases is that Web Dynpro iViews introduced with SAP NetWeaver 04s run as embedded iViews (i.e. no iFrame is used to display the iView). In chapter 0 we discuss the differences between the SAP NetWeaver 04 and SAP NetWeaver 04s releases in more detail.

Naming guideline: When we talk about Web Dynpro iViews in the following, we mean always the SAP NetWeaver **04s** version of it. In case we want to name the SAP NetWeaver 04 version we will mention this explicitly.

## Relationship to Web Dynpro iViews Created within SAP NetWeaver 04

Within SAP NetWeaver 04, you run Web Dynpro applications using an iView based on the Web Dynpro iView template. These iViews run always as *isolated iViews* within an iFrame.

In general, all created portal content based on such SAP NetWeaver 04 Web Dynpro iViews can also run on SAP NetWeaver 04s. Therefore all existing portal content based on SAP NetWeaver 04 will still run after you upgrade to SAP NetWeaver 04s.

On the other side, using SAP NetWeaver 04 Web Dynpro iViews you can not access the new features provided by the SAP NetWeaver 04s Web Dynpro iViews.

What is the relationship between these two versions?

- SAP NetWeaver **2004** Web Dynpro iViews could be added to a Web Dynpro page (i.e. a page executed by the Web Dynpro page builder) only as *isolated iViews*.
- All client-side related functionality is available for SAP NetWeaver **2004** Web Dynpro iViews running within a Web Dynpro page.
- Personalization and all services provided by the portal service factory are NOT available for SAP NetWeaver **2004** Web Dynpro iViews.
- SAP NetWeaver **2004s** Web Dynpro iViews could be added to pages executed by the PRT based page builder only as *isolated iViews*.

- All client-side related functionality is available for SAP NetWeaver **2004s** Web Dynpro iViews running on a page executed by the PRT based page builder.
- Services provided by the portal service factory are only available with some restrictions for SAP NetWeaver **2004s** Web Dynpro iViews running on a PRT page.
- Personalization is always available for NetWeaver **2004s** Web Dynpro iViews independent both running the iView on a PRT page or a Web Dynpro page.

Although the creation of Web Dynpro iViews using the Web Dynpro iView template is not disabled within SAP NetWeaver 04s we **strongly** recommend using only the SAP NetWeaver 04s Web Dynpro iViews to run Web Dynpro applications running on a SAP NetWeaver 04s installation. Using a mixture of these two version create inconsistent portal content with different behavior.

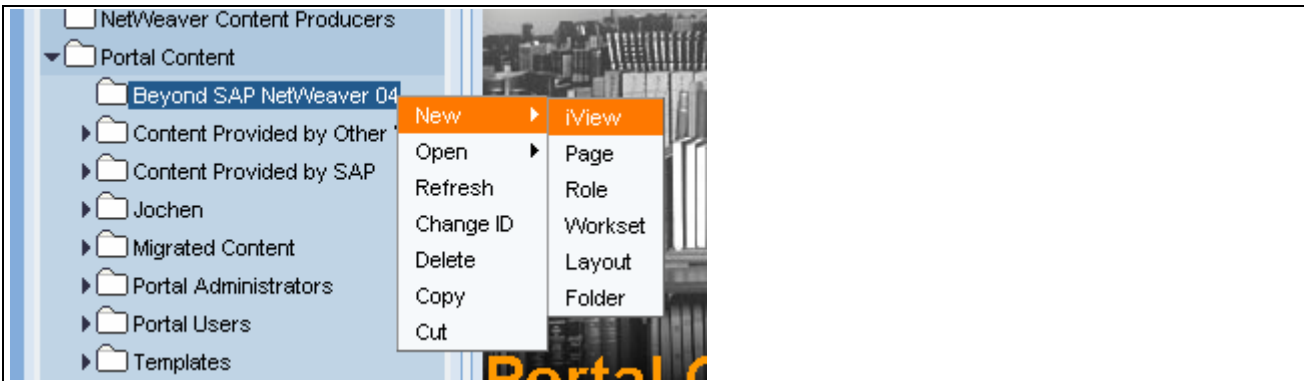
### Black Box iViews vs. Split Applications

In general, there are two types of Web Dynpro iViews within the SAP NetWeaver 04s release. First there are the *black box Web Dynpro iViews*, which from one perspective are very similar to SAP NetWeaver 2004 based iViews. The whole Web Dynpro application is rendered as a *single* (embedded) iView and the portal does not know anything about the internal structure of this Web Dynpro application.

On the other hand, one Web Dynpro application could be spit into several iViews. This makes sense for example if there is a need to provide role-based or end-user based customizations of the application layout or if you want to provide iView functionalities (like printing) for single parts of your Web Dynpro application. To enable the splitting of a Web Dynpro application into several iViews the Web Dynpro application has to follow some guidelines. We will discuss this in more detail in chapter 0.

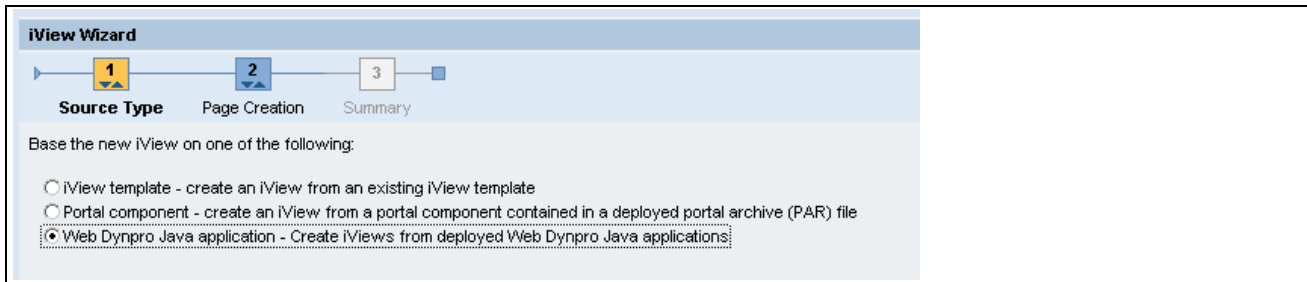
### Creating a Black Box iView

To create a black box iView you have to start the iView creation wizard by selection the context menu of the folder, which should contain the created iViews later. Please choose **New -> iView** in the context menu.



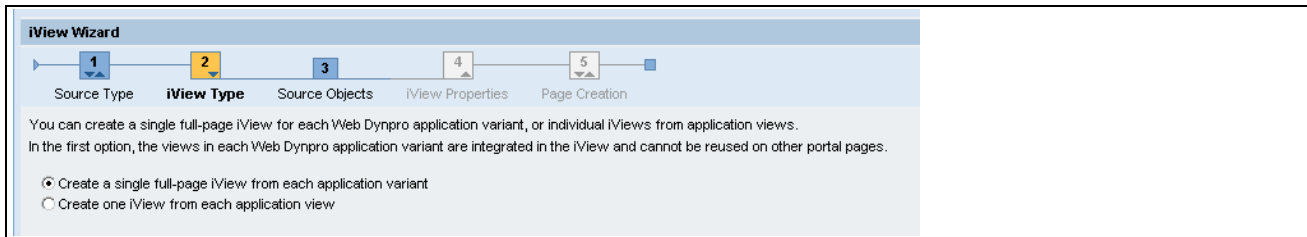
### Starting the Web Dynpro iView wizard

In the first step of the iView wizard you have to select the third option to create iViews based on a Web Dynpro (for Java) application. As mentioned earlier Web Dynpro for ABAP applications are integrated as SAP NetWeaver 04 Web Dynpro iViews based on the Web Dynpro iView template.



### Selecting the Web Dynpro application as source of the iView

On the next step you define the needed iView type. Here you have to differentiate between a black box iView and a split application. To create a black box iView you select the first option.



### Create a black box iView

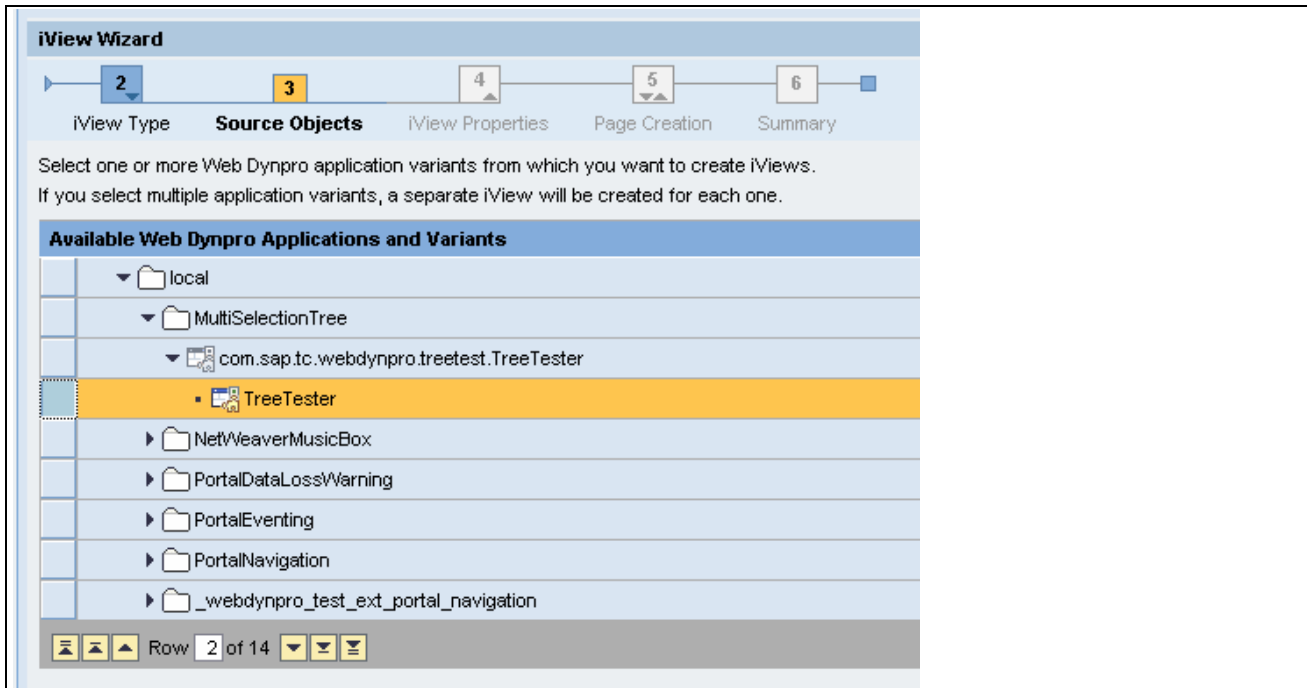
The naming here is somehow misleading. A “full-page” iView is *not* related to the iView size, you can define later one for the iView using the iView editor.

In the next step you have to select the Web Dynpro application you want to run within the SAP NetWeaver Portal. To make this selection easier you do not have to know the needed parameters like in SAP NetWeaver 04 any more, but you can drill down the deployed Web Dynpro applications.

Therefore a Web Dynpro iView is always created only for *local* Web Dynpro applications. A local Web Dynpro application is an application running on the same SAP NetWeaver 04s installation as the SAP NetWeaver Portal itself.

Using the *Federated Portal Network (FPN)* you can nevertheless separate a SAP NetWeaver Portal installation and a SAP NetWeaver installation running the used Web Dynpro applications. We will discuss this in detail in part 11 of this series.

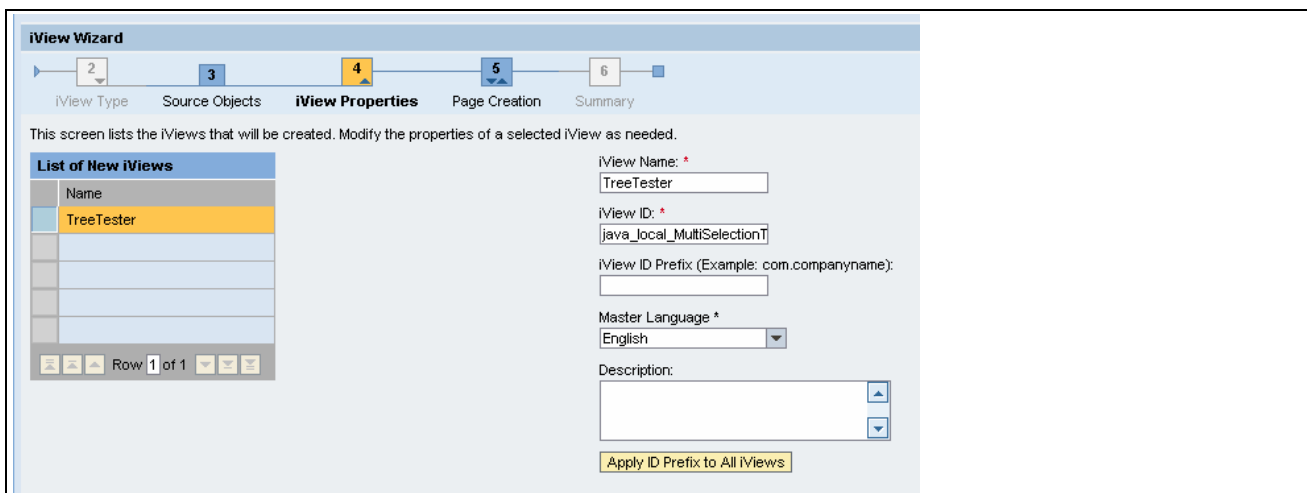
Also in this step the naming could confuse you a little bit. The mentioned *variants* of a Web Dynpro application are only relevant for SAP-internal applications (especially applications, which are part of myERP 2005). All other applications exposed only one single variant, which is named as the application itself. The next figure shows as an example the selection of the *TreeTester* application.



### Selecting the Web Dynpro application

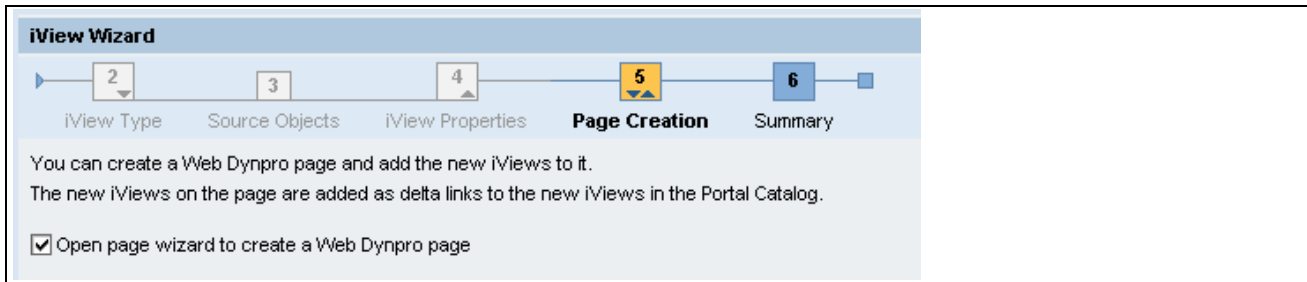
The displayed hierarchy shows in the first level the vendors of the deployed Web Dynpro development components and / or Web Dynpro projects. The second level visualizes the names of the deployed Web Dynpro development components and / or Web Dynpro projects. The third level shows the Web Dynpro applications of the selected development component or project. Depending on the used Web Dynpro iView type the fourth level shows the variant(s) or the list of exposed iViews (as described above).

After selecting the Web Dynpro application you can modify in the next wizard step some generic iView settings like the iView name or the technical iView ID if needed. Normally the proposed values could be used without any changes.



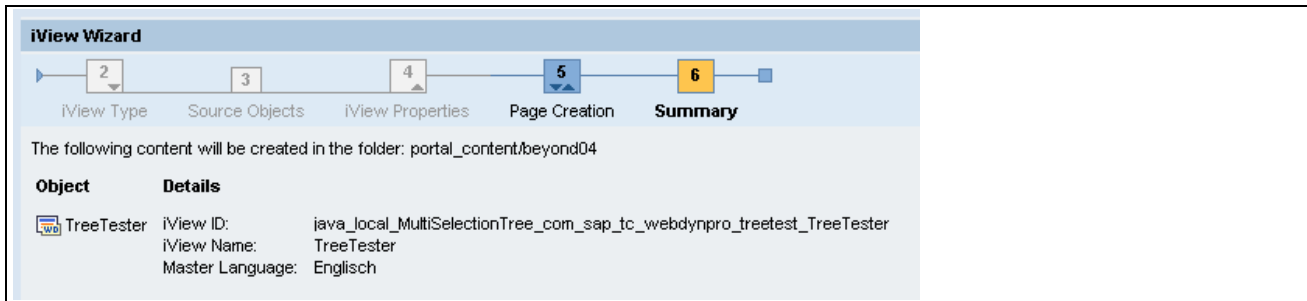
### Defining basic iView properties

In the next step you can optionally create a Web Dynpro page containing the created black box iView. If you want to create such a Web Dynpro page you have to select the **Open Page wizard to create a Web Dynpro page** checkbox. The single steps to create a page during the creation of an iView are the same steps as described above.



### Creating a Web Dynpro page

In the last step you get a summary of the created portal objects, i.e. the iView(s) and optional also the Web Dynpro page. After pressing the **Finish** button the objects are created.



### Summary of created objects

As you can see in the figure below, also the Web Dynpro iView is visualized in a specific way. Please keep in mind that this visualization is only used for the SAP NetWeaver 04s version of a Web Dynpro iView. The SAP NetWeaver 04 version is visualized using the standard iView icon.

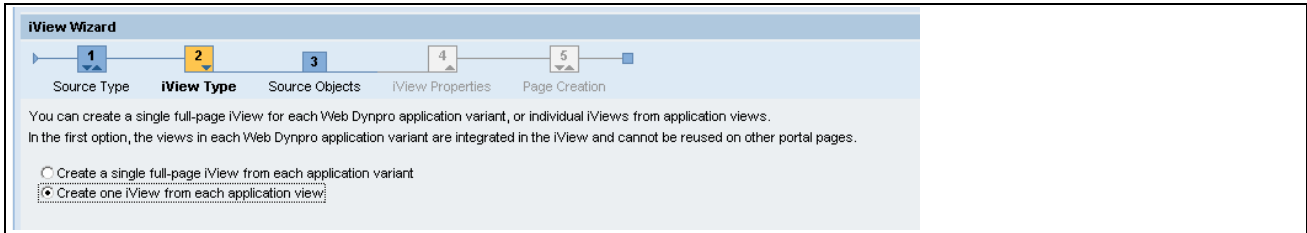


### Visualization of a Web Dynpro iView

#### Defining Several iViews Based on One Web Dynpro Application

To create more than one iView for one Web Dynpro application you have to follow more or less the same flow than during the creation of a black box iView.

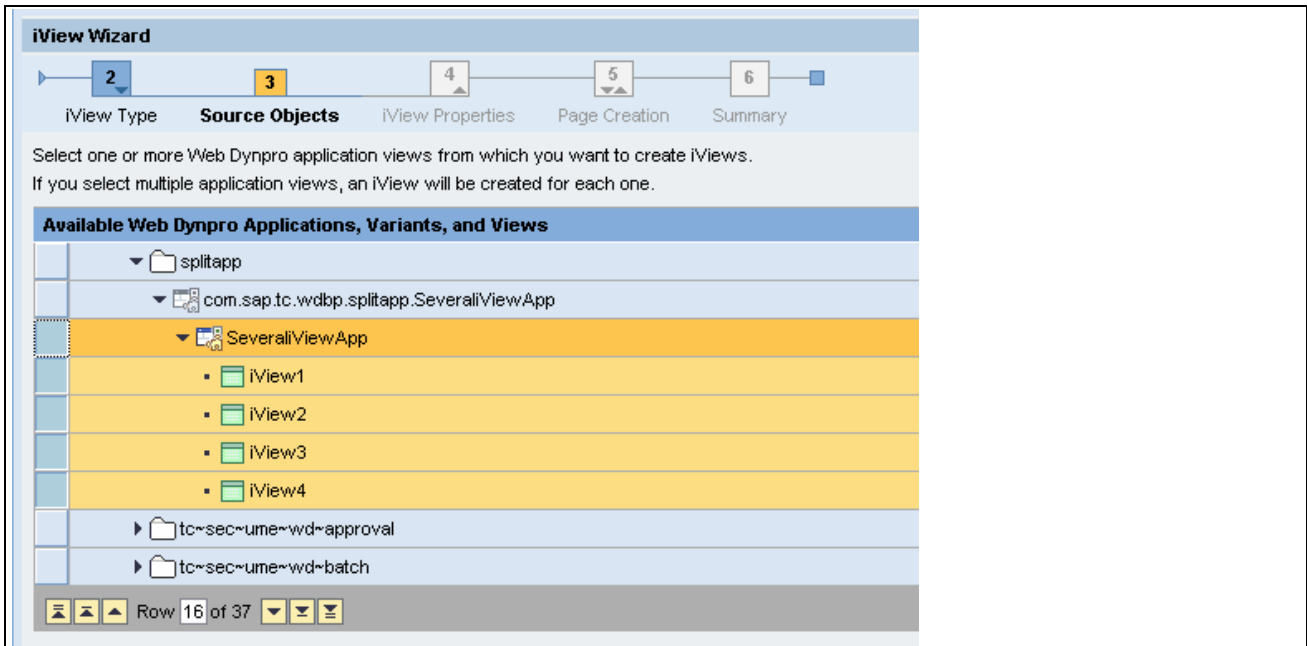
In the second step you have to select the second option to create several iViews for one Web Dynpro application. Again the naming could be somehow misleading: the defined iViews by a Web Dynpro application are not necessarily related to Web Dynpro views. We will discuss this in detail later in the document.



### Creating several iViews for one application

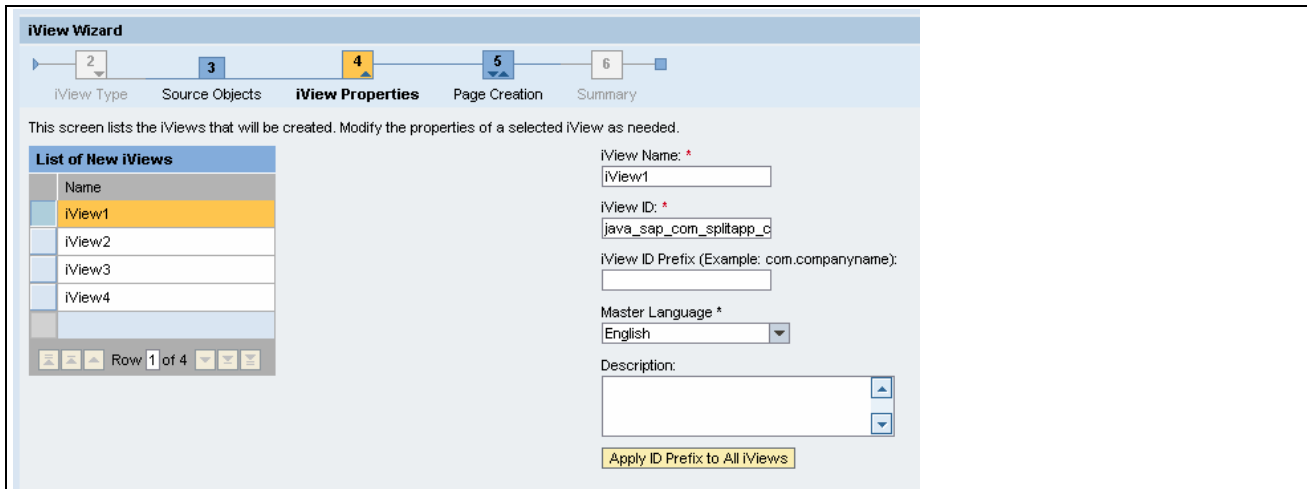
In the next step you select instead of an application variant the iViews you want to create for the Web Dynpro application. Selecting the variant automatically selects also all iViews provided by this variant.

In the shown example we create for the `SeveraliViewApp` application the iViews `iView1`, `iView2`, `iView3` and `iView4`. If you want to create a Web Dynpro application exposing more than one iView you have to follow some guidelines – we describe this in detail in the next chapter.



### Selecting all iViews of an application

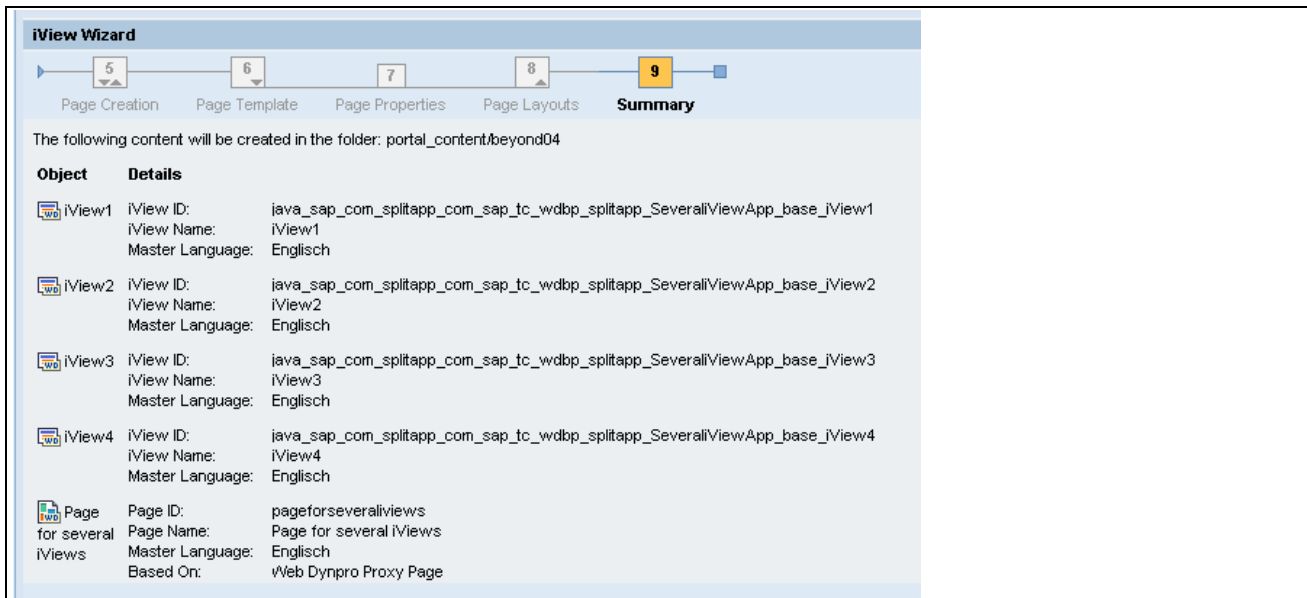
After selecting the needed iViews you can define in the next step some basis iView properties. As for the creation of one single iView for an application you could just accept the proposed values.



### Defining basic iView properties

When you create several iViews of one Web Dynpro application it often makes sense to create also a Web Dynpro page containing all these iViews. You can do this as described in the last chapter also within the iView creation wizard.

Before creating all needed objects you get in the last step a summary of the objects like displayed in the following figure.



### Summary of the created objects

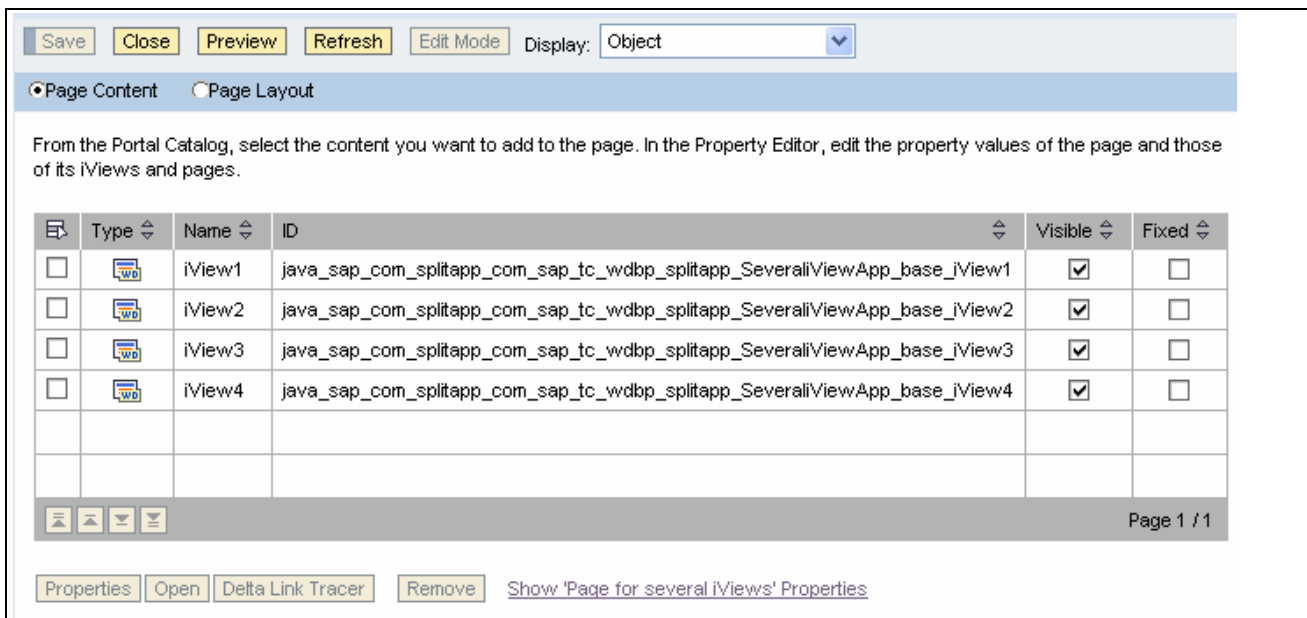
After pressing the **Finish** button all objects are created and stored in the selected folder in the content catalog. Besides the created iViews *iView1*, *iView2*, *iView3* or *iView4* also the *Page for several iViews* Web Dynpro page was created containing these iViews.

In general you can add the created iViews of course also to an existing Web Dynpro page. But often it makes sense to create such a page together with the iViews.



*Created iViews in the content catalog*

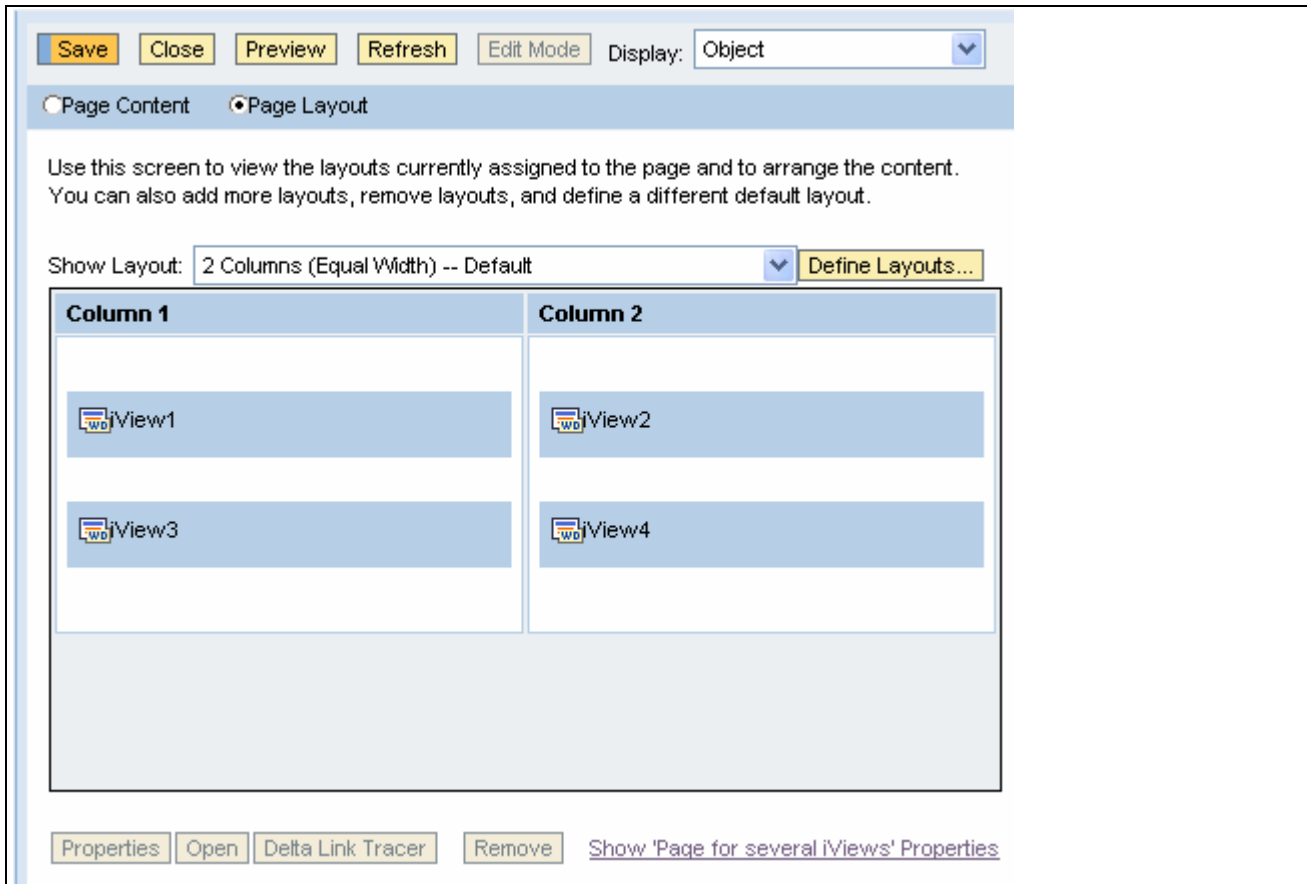
To have a deeper look into the created page you open the page editor using an **Open -> Object** context menu entry of the **Page for several iViews** page. The page editor provides two different views to the content of the page – the **Page Content** shows the iViews, which are part of this page. The following figure shows the four iViews we created before.



*Content of the Page for server iViews page*

As the Web Dynpro page is in the end a standard page you could add any other (Web Dynpro) iView to this page.

Choosing the **Page Layout** radio button you switch to the layout of the page. You can define any layout you want. If needed you could add also new layouts to the list of available layouts using the **Define Layouts ...** button.



### *Defining the page layout*

Using the Preview button you can start the page directly. As a result you see the four iViews defined for the page. Please keep in mind that each one runs as embedded iView. As a consequence the whole page is refreshed as soon as the user triggers any action inside one of the Web Dynpro iViews. As the Web Dynpro page itself is rendered by the Web Dynpro page builder, the whole page behaves as any other Web Dynpro application, i.e. the reload of the page is done in a flicker-free mode.



### *Running the Page for several iViews page*

After describing the creation of one or more iViews for one Web Dynpro application we will discuss in the next chapter how to implement a Web Dynpro application exposing several iViews.

### **Exposing a Static List of iViews**

Splitting up one Web Dynpro application into several iViews allows you to reuse all existing portal functionality to layout these iViews in different ways. Also the end user could re-arrange the iViews using the standard page personalization. Besides the flexible layout possibilities you can provide iView specific features like printing an iView to specific parts of the whole application.

As the iViews run with the same Web Dynpro application (instance), you have on the server-side a tight integration between these iViews (i.e. between the different parts of the application, which are exposed as different iViews). Instead of using client-side mechanism like portal eventing to communicate between these iViews you could use all Web Dynpro features like context-sharing.

With NetWeaver 2004s the portal semantics like an iView or a page are not part of the Web Dynpro meta-model. Therefore there are some conventions how to define these parts, which should be exposed as different iViews in the end.

In the following we will explain these conventions using the very simple application [SeveraliViewsApp](#).

### **Using the SeveraliViewsApp Example Application**

Instead of using a standard Web Dynpro project we define our [SeveraliViewsApp](#) example application in a local development component (DC) named [LocalDevelopment~splitapp~sap.com](#).

In general we do *not* recommend to use simple Web Dynpro projects any longer – as soon as you want for example to embed Web Dynpro components stored in other development component this would not be possible for Web Dynpro projects.

[Download the local DC \(LocalDevelopment~splitapp~sap.com\) here.](#)

To get the contained Web Dynpro application [SeveraliViewsApp](#) running, proceed as follows:

1. Extracting the local Web Dynpro DC into the development workspace

Unzip the complete contents of the zip file [LocalDevelopment~splitapp~sap.com.zip](#) into your local development directory `.dtc` wherein all local DCs are stored.

Example: Your local development folder comprising all local DCs is `C:\Documents and Settings\\.dte\LocalDevelopment\DCs\sap.com`. Unzip the zip file `LocalDevelopment~splitapp-sap.com.zip` into this folder.

2. Importing the local Web Dynpro DC into the SAP NetWeaver Developer Studio
3. Start the SAP NetWeaver Developer Studio.
4. Open the Development Configurations perspective **Window -> Open Perspective -> ... Development Configurations**.
5. Select the view **Local DC** and open the **node Local Development -> My Components**.
6. Refresh the perspective **view Local DCs** by pressing the button **Refresh tree** within the title bar.
7. Select the DC `splitapp`.
8. Open the context menu of the selected nodes and choose **Build ....**
9. In the **Build Development Component** window mark all listed development components, uncheck the check box. **Only build if necessary** and choose **OK**.
10. Select the DC `splitapp` again.
11. Open the context menu of the selected node and choose **Create Project**.
12. Open the Web Dynpro perspective Window -> Open Perspective -> ... Web Dynpro.

The Web Dynpro DC `LocalDevelopment~splitapp-sap.com` will appear in the **Web Dynpro Explorer** view.

Depending on which Java compiler preferences are set, you may see some warnings in the "Task" view after importing the project.

If the severity level for problems of type "Unused imports" (set in Preferences – Java – Compiler) has the value "Warning", the compiler will issue a warning for unused import references.

Ignore these warnings!

13. Deploying and running the Web Dynpro application `SeveraliViewsApp`.

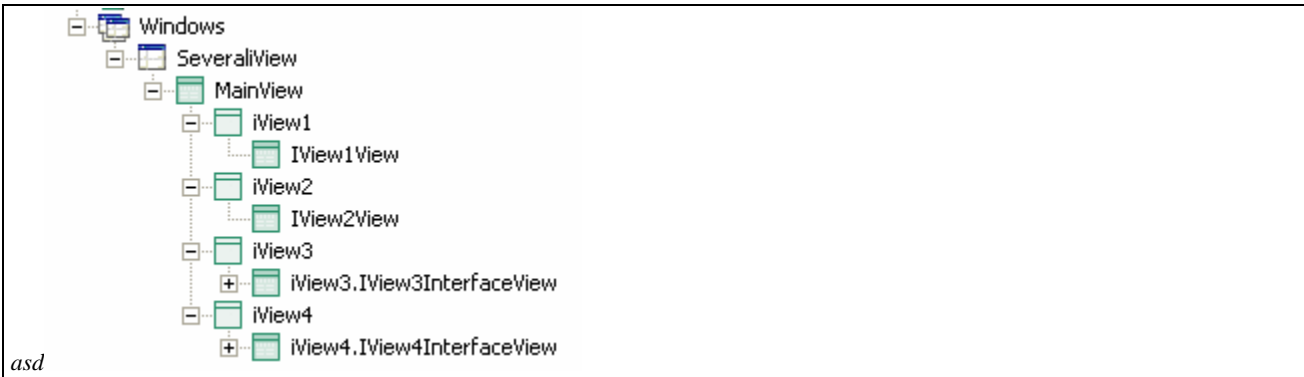
## Defining the iViews

As written in the last chapter portal semantics like an iView or a page is not part of the Web Dynpro meta-model. Therefore we define the list of exposed iViews using some simple conventions:

- Each iView is defined by a `ViewContainerUIElement` UI element defined in the root view of the application window.
- This root view could define also other UI elements around the `ViewContainerUIElement` UI elements – but these UI elements are NOT visible if the different iViews are rendered in the portal. Only the content of the `ViewContainerUIElement` UI elements are visible in the end as different iViews.
- The used `ViewContainerUIElement` UI elements should define meaningful IDs, because these IDs are used in the end as the default iView titles.
- The content of each `ViewContainerUI` UI element could be defined both using a simple Web Dynpro view and an interface view of an embedded Web Dynpro component. Using the interface view you can implement the exposed iViews but different Web Dynpro components.

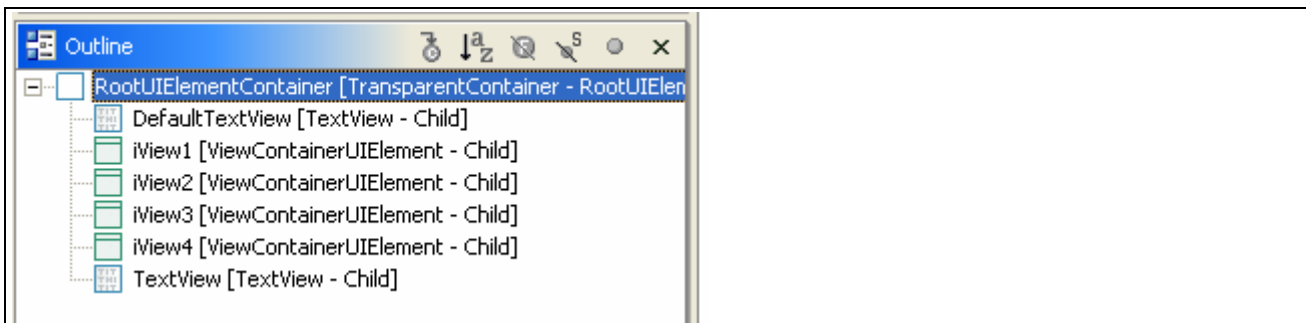
In the following we will discuss these basic conventions using the `SeveraliViewsApp` example application.

The following figure shows the view assembly of the example application. The `SeveraliViews` window contains the `MainView` Web Dynpro view acting as the root view. As described above this view the list of exposed iViews. In the showed example the iViews `iView1`, `iView2`, `iView3` and `iView4`.



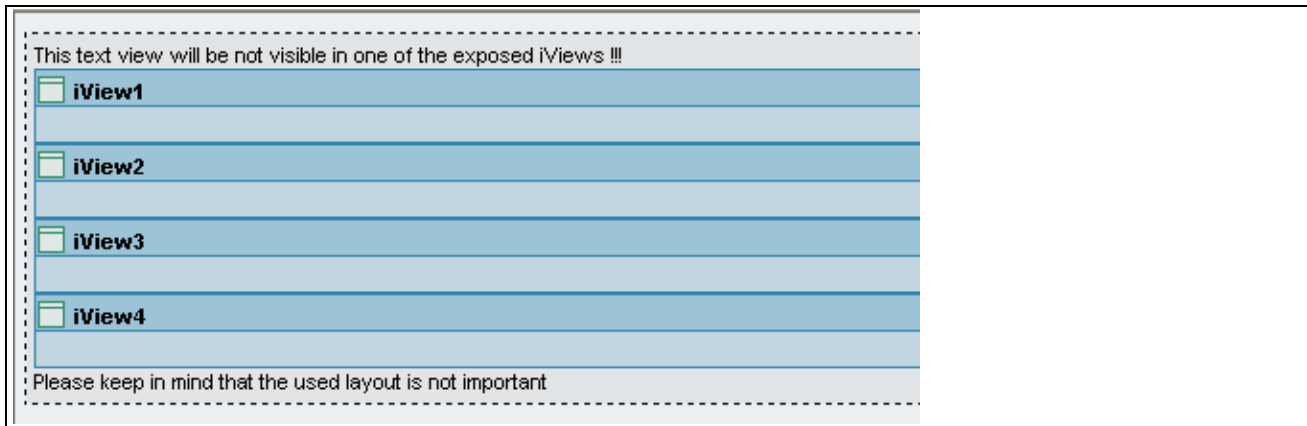
### *Embedding of views and interface views*

The next figure shows the definition of these iViews using several instances of `ViewContainerUIElement` UI elements. The title of an iView is defined by the ID of the associated `ViewContainerUIElement` UI element.



### *Declared list of exposed iViews*

The main task of the `MainView` view is to define the exposed iViews. The used layout is not relevant for that. Only in case you would like to start the application standalone (i.e. not running in the portal) you could define a useful layout for the `MainView` view.



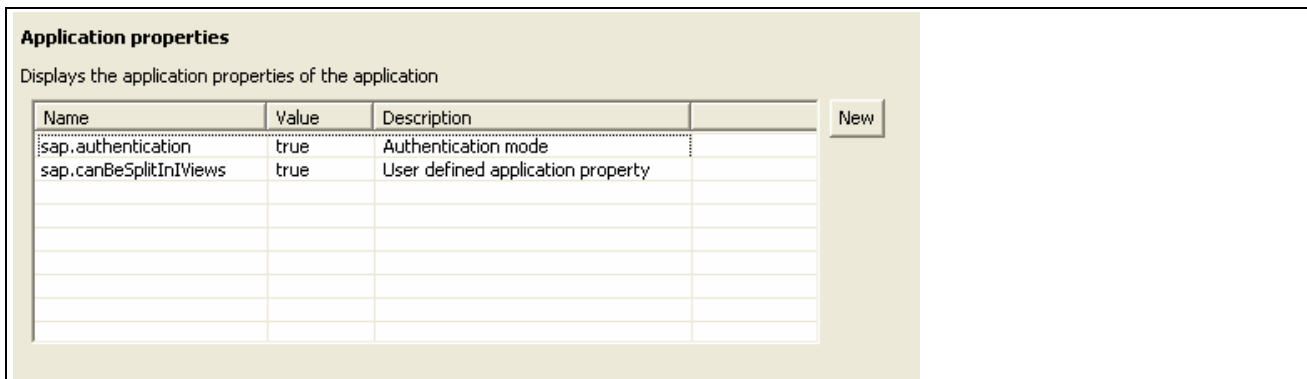
*Visual representation of the root view*

When running the several iViews in the portal only the content of the `ViewContainerUIElement` UI elements defining the iView are displayed. The text views for example, which are added to the `MainView` view, are not visible in the portal in the end.

Regarding the defined content of a `ViewContainerUIElement` UI element you have the same option as for any other `ViewContainerUIElement` UI element. You can embed another Web Dynpro view or you could use a component interface view of another embedded component. Especially the second option is a very sophisticated way to implement the different iViews using different components.

In the example application the `iView1` and `iView2` iViews are implemented using additional Web Dynpro views `iView1` and `iView2`. The `iView3` and `iView4` are implemented by using the interface views `iView3.IView3InterfaceView` and `iView4.IView4InterfaceView`.

Besides these guidelines you have to mark an application explicitly as an application defining several iViews by defining the `sap.canBeSplitInIViews` application property as shown in the next figure.



*Definition of the `sap.canBeSplitInIViews` property*

## Launching a Web Dynpro iView

After discussion the creation of Web Dynpro based portal content, we would like to explain a little bit more in detail, what happens when starting a Web Dynpro iView.

As mentioned above the Web Dynpro page builder is responsible to start the Web Dynpro application launched as Web Dynpro iView (please keep in mind that we talk here about the SAP NetWeaver 04s version). On the other side runs the Web Dynpro page builder itself also as a standard Web Dynpro application. Launching this application is under the responsibility of the *SAP Application Integrator*.

What happens if a Web Dynpro iView is launched?

The iView URL is executed by the Portal Runtime (PRT) and the SAP Application Integrator is called. The SAP Application Integrator is implemented as a portal component. An URL to a Web Dynpro iView looks like this:

```
http://<host>.<domain>:<port>/irj/servlet/prt/portal/prtroot/com.sap.portal.pagebuilder.IviewModeProxy?iview_id=pcd:portal_content/beyond04/beyondw04role/pageforseveraliview/java_sap_com_splitapp_com_sap_tc_wdbp_splitapp_SeveralIViewApp_base_iView1&iview_mode=default
```

The `iview_id` parameter defines the Web Dynpro iView, which should be launched.

The main task of the SAP Application Integrator is to start the Web Dynpro page builder application instance. To do this the SAP Application Integrator computes the needed Web Dynpro application URL pointing to the page builder instance and passes some further information about several portal settings (like the used theme or the current portal language). The computed URL looks like this:

```
http://<host>.<domain>:<port>/webdynpro/dispatcher/sap.com/pb/PageBuilder
```

Obviously this is a standard Web Dynpro application URL to launch the `sap.com/PageBuilder` application.

Within the SAP NetWeaver 04s release the SAP Application Integrator posts always the parameters. To check, which parameters are passed to the page builder application you can use a tool like *HttpWatch*. The following figure shows these parameters using *HttpWatch*.

Parameter	Value
sap-ext-sid	dpxsBRLQ5ml9xAA508v4+A==8tUCB28oHvJc55GggRuhw==
sap-wd-ctwridid	WID1146813754400
sap-wd-tstamp	1146813754430
PagePath	pcd:portal_content/beyond04/beyondw04role/pageforseveraliview/java_sap_com_splitapp_com_sap_tc_wdbp_splitapp_SeveralIViewApp_base_iView1
sap-wd-app-namespace	zzzz
sap-ep-version	7.00.200604252130
sap-locale	en
sap-accessibility	
sap-rtl	
sap-cssurl	http://wdfid00165826a.wdf.sap.corp:53000/irj/portalapps/com.sap.portal.design.urdesigndata/themes/portal/sap_tradeshaw/ur_jur_ie6.css?7.0.5.0.0
sap-cssversion	7.0.7.0.0
com.sap.portal.reserved_wd_pb_restart	false
DynamicParameter	

### List of passed parameters

Besides the parameters defining the needed portal settings like `sap-cssurl` or `sap-ep-version` the called Web Dynpro iView is passed using the `PagePath` parameter (this is somehow misleading: also in case that one single iView is launched the parameter is called `PagePath`!).

The Web Dynpro page builder uses this path to lookup the needed iView properties, starts the associated Web Dynpro application and forwards the portal settings to it.

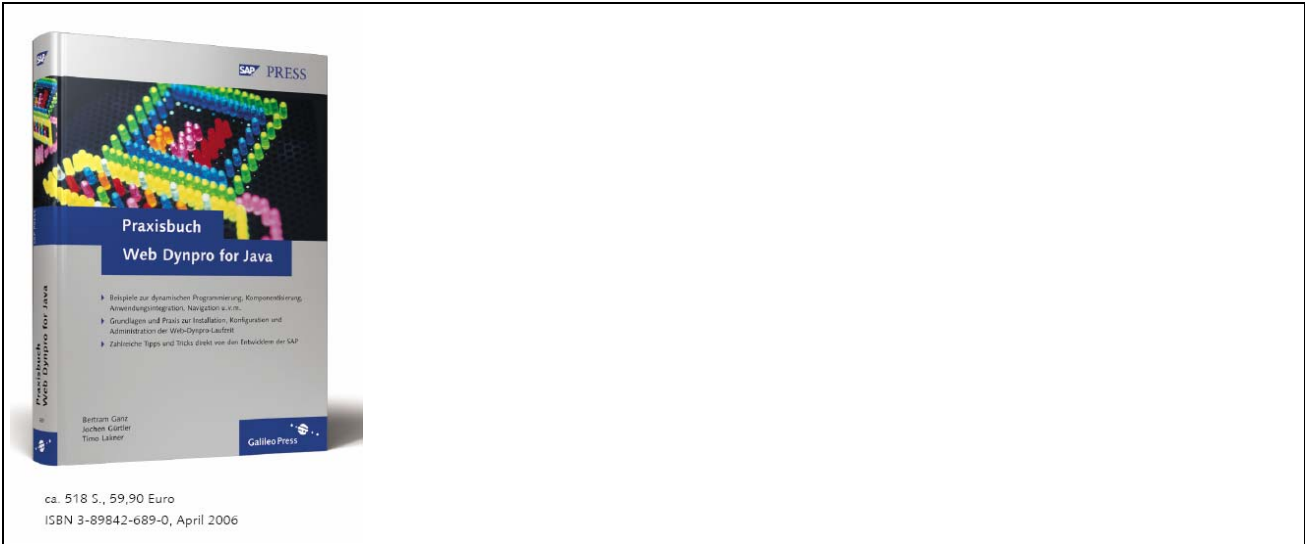
To launch a Web Dynpro page more or less the same steps are executed. Only the last step looks a little bit different. Instead of running exactly one single Web Dynpro application when launching a Web Dynpro iView, the Web Dynpro page builder has to run all iViews of the Web Dynpro page instead. Besides that the page builder has to layout them in the configured way.

In the end of this chapter we would like to shortly discuss some consequences of the described flow.

First you have to keep in mind that using the Web Dynpro page builder creates always one additional Web Dynpro application instance. If you launch for example *one* Web Dynpro iView you run *two* Web Dynpro applications: one for the page builder itself and one for the called Web Dynpro application. If you run *one* Web Dynpro page containing *n* Web Dynpro iViews you run *n+1* application instances. To avoid time-consuming recreations of the needed page builder application instance every time, a user navigates from one Web Dynpro iView / page to another Web Dynpro iView / page, the page builder application instance is reused over the whole user session.

## Further Information

To get more information especially about extended features of the SAP Application Integrator we would like to bring to your attention to a new Web Dynpro for Java book. The German version *Praxisbuch Web Dynpro for Java* is available now – the English translation *Maximizing Web Dynpro for Java* will follow soon.



### *Praxisbuch Web Dynpro for Java*

“Maximizing Web Dynpro for Java” is aimed at Web Dynpro developers who already have development experience but want to go to the next level. As members of the Web Dynpro development team, the authors of this book are expert at revealing tips and tricks for avoiding inefficiencies and pitfalls. They cover many areas including the SAP NetWeaver Development Infrastructure, component-oriented application design, portal integration, object-based navigation, installation tips, configuration and administration, and more. There are also complete examples on how to write web service or portal-based applications.

Note: This book is currently available in German only. The English version will be released in August/September 2006. You can order the book through [SAP](#) or [Amazon.com](#).



### *Maximizing Web Dynpro for Java*

## 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.