Integrating Web Dynpro and SAP NetWeaver Portal: Personalization of Web Dynpro Java applications

Applies to:

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

Summary

This article describes the available personalization capabilities when running Web Dynpro Java applications as Web Dynpro iView within the SAP NetWeaver Portal. This includes the relationship to the portal personalization, the implicit Web Dynpro personalization and the explicit personalization of iViews.

Author: Jochen Guertler

Company: SAP AG

Created on: 05th February 2007

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 “Maximizing Web Dynpro for Java” book.

Table of Contents

Introduction

Running Web Dynpro applications as Web Dynpro iViews within the SAP NetWeaver portal provides a wide range of additional functionally. One of the most powerful mechanisms is the usage of the different flavors of personalization. The goal of this document is to give an comprehensive overview of the available capabilities and their relationship to Web Dynpro applications.

Classification of personalization

Before we describe the concrete usage of personalization features within a Web Dynpro application we would like to categorize the different flavors available in the SAP NetWeaver portal.

As personalization is used for many more or less different things – we would like to define some naming conventions also.

In general we differentiate between the following personalization capabilities:

- Portal personalization

  With *portal personalization* we mean the possibility to maintain several global portal settings like the used theme or the used language. These settings are true for all running portal content independently of the concrete content type. We will describe below if and how these global settings are used when running a Web Dynpro iView.

- Explicit iView personalization

  The *explicit iView personalization* defines the capability to adjust iView properties which are explicitly defined for a specific iView. These iView properties could be adjusted for one single user or for a group of users. We will describe later how to define and access such iView properties within a Web Dynpro application.

- Implicit Web Dynpro personalization

  The *implicit Web Dynpro personalization* allows customizing of visual aspects of Web Dynpro UIs without any additional programmatically effort or configuration. Typical examples are hiding UI elements or changing labels or tool tips. Using the implicit Web Dynpro personalization you can adjust Web Dynpro UIs both for one single user and for a group of users.

Single user vs. group of users

Besides the different flavors of personalization as described in the last chapter we have to differentiate between two different scopes for a personalization. The natural scope of personalization is obviously the *single user*. For this scope all personalization changes are only visible for the user defining the changes. As this is typically called *end-user personalization* we will use this also in the following document.

Besides that there is also the need to adjust a UI or specific properties for a *group of users*. This group of users is typical defined by a *role*. Therefore the personalization for a group of users is sometimes named *role-based personalization* (although this sounds a little bit contradictory) or *role-based customization* (although customization is also used in totally different areas).

In this document we will use *personalization for a group of users* as this defines it best from our point of view.

Classification of Web Dynpro iViews

Before we continue with the different flavors of the personalization we have to clarify the naming used in this document regarding the different types of Web Dynpro iViews.

We differentiate between two types of Web Dynpro iViews. First there are Web Dynpro iViews based on the *Web Dynpro iView template*. This type of Web Dynpro iView could only be run as isolated iViews. As this iView type is available since the SAP NetWeaver 04 release we name this *NW04 Web Dynpro iView*.

Second there are Web Dynpro iViews based on the *Web Dynpro page builder*. This type of Web Dynpro iView could be run embedded in Web Dynpro pages. As this iView type is available since the SAP NetWeaver 04s release we call this *NW04s Web Dynpro iView*.

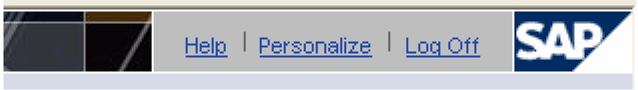
For more details about the differences and capabilities of the Web Dynpro iView types please check the related document of this series.

After this general classification and naming we start in the next chapter describing the portal personalization more in detail.

Portal personalization

The portal personalization is a generic feature of the SAP NetWeaver portal. From a Web Dynpro application point of view there is nothing to do to enable it.

To enable the portal personalization for a specific user you have to assign the employee role to this user. Doing this the user gets a **Personalize** link in the portal header as shown in the next picture.



Provided settings

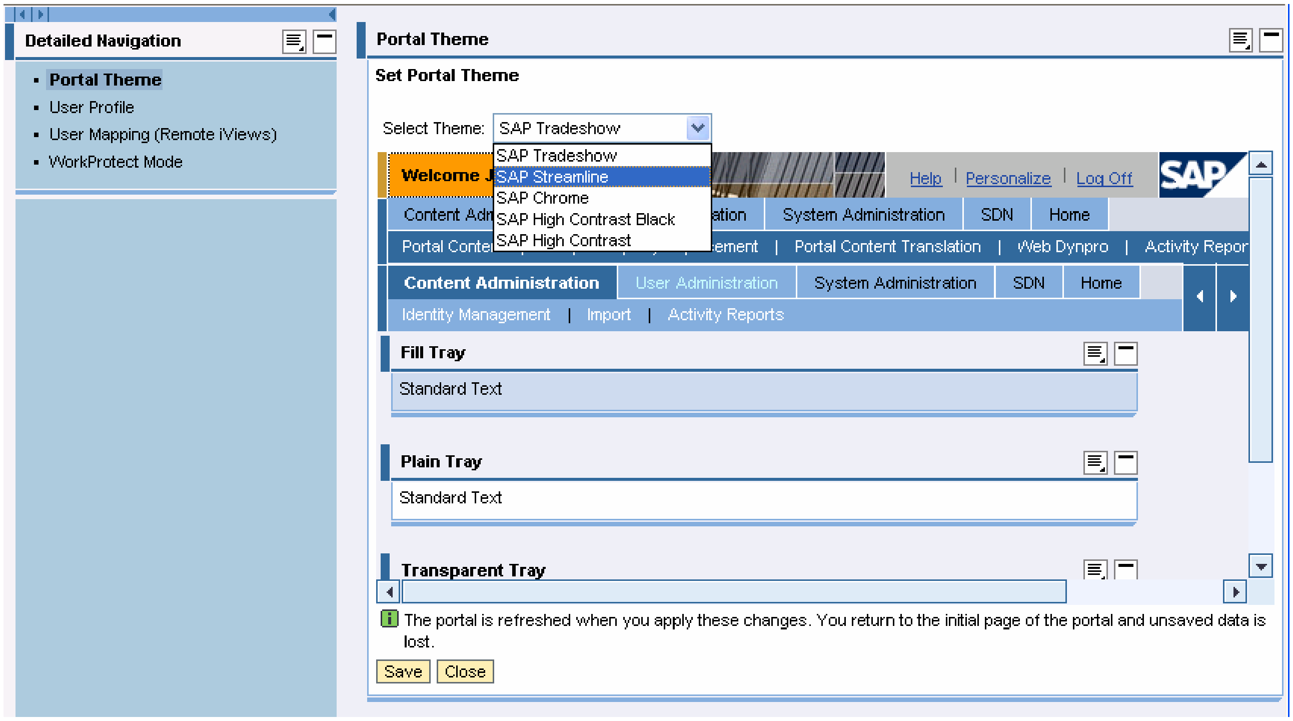
Using this link the user could open the portal personalization dialog providing different settings:

- Portal Theme
- User Profile
- User Mapping (Remote iViews)
- Work Protect Mode

We will discuss these settings in the next chapters more in detail.

Portal Theme

The **Portal Theme** setting allows defining the used portal theme. The portal personalization dialog provided a list of themes assigned to the current portal desktop as shown on the next picture.



Select Theme: SAP Tradeshow

The portal is refreshed when you apply these changes. You return to the initial page of the portal and unsaved data is lost.

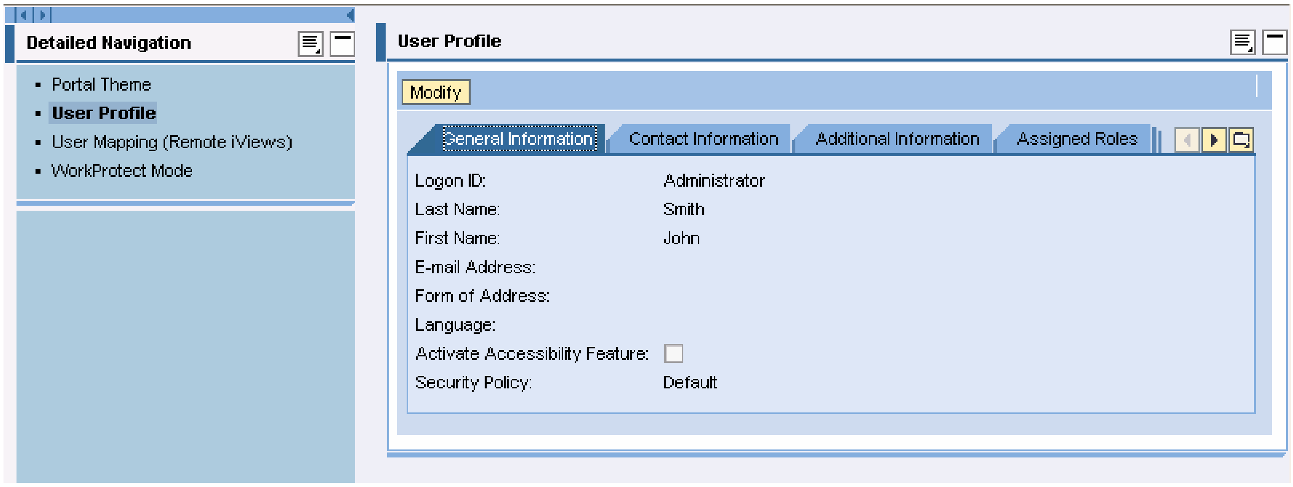
To get more information how to assign a certain theme to a portal desktop please check the related documents under sdn.sap.com or help.sap.com.

A Web Dynpro application running as Web Dynpro iView is automatically using the current portal theme. Only in case that this portal theme is not compatible to the theme needed by the Web Dynpro runtime a fallback theme is used.

To get more info about the theme handling of Web Dynpro applications please check the related document of this series.

User Profile

The User Profile setting provides access to the user profile, i.e. all user related data like the simple one like first name or the e-mail address or also more complex one like the assigned roles. If you would like to change some of the data you have to switch to the edit mode using the **Modify** button. All changes are stored persistently and are available even after a system restart.



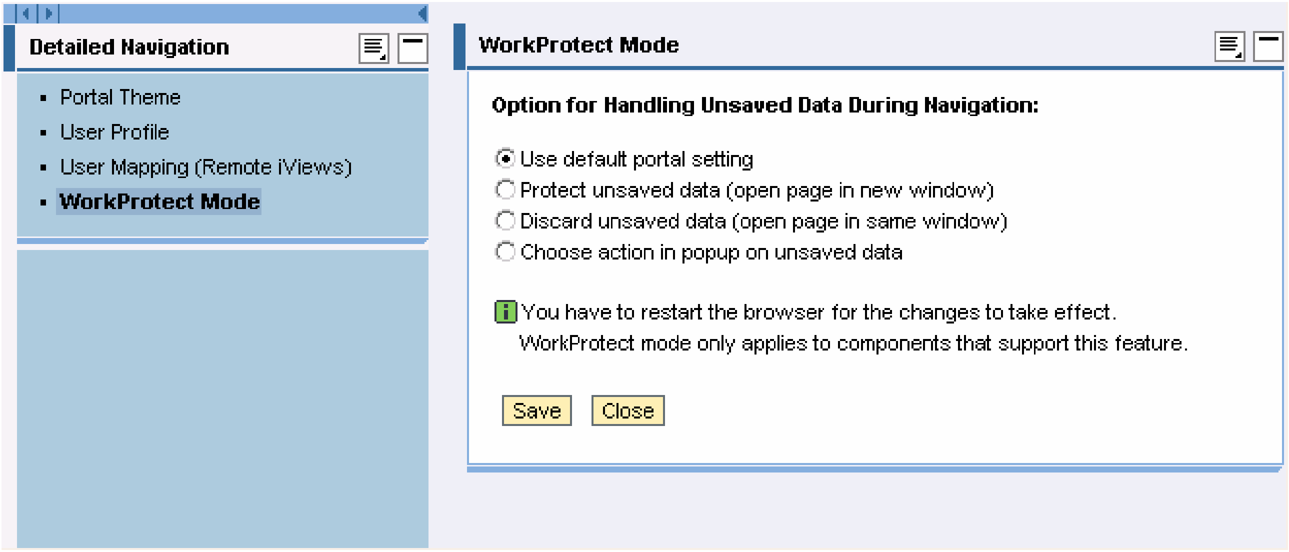
Logon ID:	Administrator
Last Name:	Smith
First Name:	John
E-mail Address:	
Form of Address:	
Language:	
Activate Accessibility Feature:	□
Security Policy:	Default

Within a Web Dynpro application you have access to most of the data stored in the user profile using the `WDClientUser` class.

Work Protect Mode

Using the **Work Protect Mode** setting you could specify the expected behavior in case that there is unsaved data and the user closes the browser or navigates to another iView.

Please check the portal documentation under sdn.sap.com or help.sap.com to get more details about that.



Option for Handling Unsaved Data During Navigation:

● Use default portal setting
○ Protect unsaved data (open page in new window)
○ Discard unsaved data (open page in same window)
○ Choose action in popup on unsaved data

You have to restart the browser for the changes to take effect.
WorkProtect mode only applies to components that support this feature.

You could use the portal Work Protect Mode using the `WDPortalWorkProtectMode` class. The settings done in the portal personalization dialog are in any case completely transparent for a Web Dynpro application.

After the discussion of the capabilities of the portal personalization we will now get into the details about the explicit personalization of iViews

Explicit iView personalization

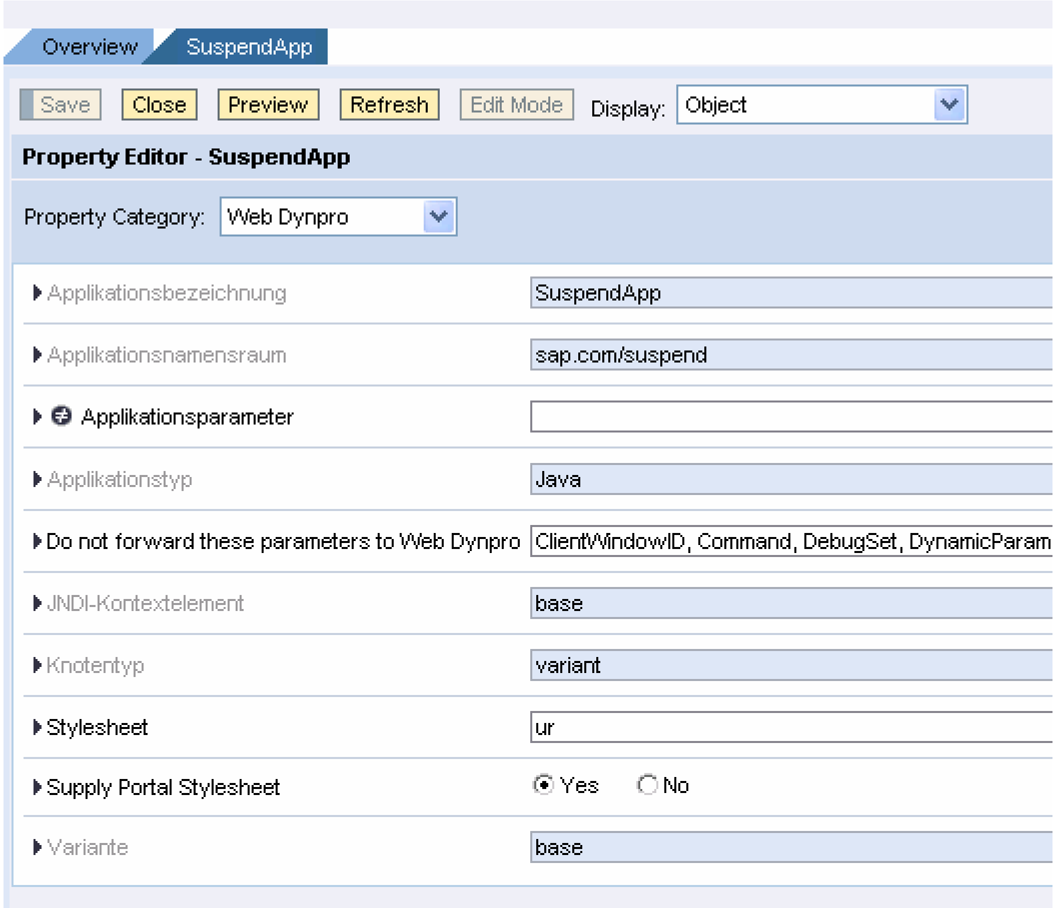
Modeling portal content

One of the basic concepts of the SAP NetWeaver portal is the *portal content model* defining a relatively small set of objects used to model all kind of running content. The most prominent objects are *iViews*, *pages* or *roles* used to define the running applications and their layout and structure inside the SAP NetWeaver portal.

One common capability of all these objects is a list of properties defining the different aspects of an object. This list of properties consists of a set of generic properties available for all kind of objects and additional object-specific properties. As the list of properties could be really long, all properties are grouped in *categories*.

All properties could be maintained by a content administrator in the *property editor*. To start the property editor for a specific object you have to browse to the object under **Content Administration -> Portal Content** and open it using the **Open Object** entry in the object-specific context menu.

The following picture shows the property editor opened for a Web Dynpro iView. As described above there are several categories available for an object. In the shown picture the **Web Dynpro** category is selected, containing all Web Dynpro specific iView properties.



Display: Object

Property Editor - SuspendApp

Property Category: Web Dynpro

Applikationsbezeichnung	SuspendApp
Applikationsnamensraum	sap.com/suspend
Applikationsparameter	
Applikationstyp	Java
Do not forward these parameters to Web Dynpro	ClientWindowID, Command, DebugSet, DynamicParam
JNDI-Kontextelement	base
Knotentyp	variant
Stylesheet	ur
Supply Portal Stylesheet	● Yes ○ No
Variante	base

As you can see in the picture above only some of the shown properties are editable. Some of the properties are read-only. Technical information like the application name or the name of the Web Dynpro development component containing the application is for obvious reasons not editable.

Defining iView properties

The definition of application specific iView properties within the SAP NetWeaver 04s release is *not* supported both for NW04 and NW04s Web Dynpro iViews.

The only way to use the explicit iView personalization for application specific properties is the usage of the generic `Application Parameters` iView property, which is available both for NW04 and NW04s Web Dynpro iViews. Using this property it is possible to pass several parameters to an Web Dynpro application. A typical value looks like `customerID=4711&display=large&visibleRows=10`.

The SAP Application integrator adds this parameter string to the URL starting the Web Dynpro application. To access these parameters within a Web Dynpro application you have to use the `WDProtocolAdapter` class as shown in the following code example:

```
String customerID =
      WDProtocolAdapter().
      getProtocolAdapter().
      getRequestObject().
      getParameter(“customerID”);
```

Keep in mind that URL parameters are *not* forwarded to all follow-up requests. You have to access this parameter therefore in the initial request (e.g. in one of the `wdDoInit()` methods of a component or custom controller).

Accessing iView properties within a Web Dynpro application

Although there is currently no possibility to define application-specific iView properties it could make sense to access generic iView properties within the Web Dynpro application. We will describe below one example for this. Depending on the used Web Dynpro iView type there are different solutions doing this.

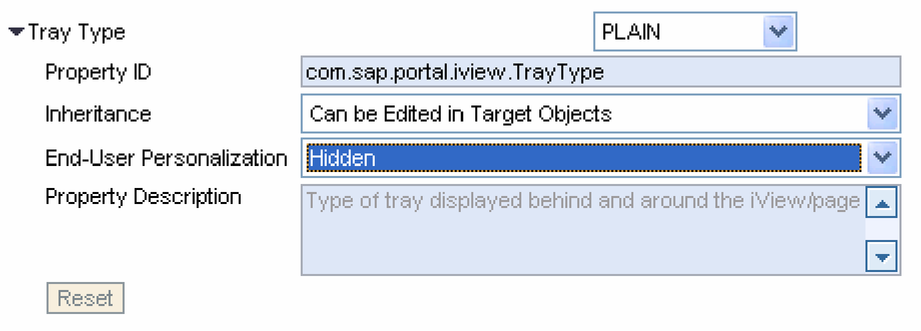
Using NW04 Web Dynpro iViews

To access iView properties within a Web Dynpro application running as a NW04 Web Dynpro iView the first step is to inform the SAP Application Integrator which iView properties have to be forwarded to the Web Dynpro application. Please keep in mind that NW04 Web Dynpro iViews are completely isolated from the SAP NetWeaver portal. Therefore the only way to pass parameters from the SAP NetWeaver portal to the Web Dynpro runtime is using the URL calling the Web Dynpro application.

For this use case the SAP Application Integrator provides several variables which are resolved during the computation of the Web Dynpro application URL. To pass a specific iView property (better: the value of it) we have to use the `<Profile.xyz>` variable. `xyz` defines here the technical ID of the iView property.

Please have a look at the following example: we would like to pass the `Tray Type` iView property to a Web Dynpro application. This could be used to adjust the background color of the Web Dynpro application based on the defined tray type of the iView running the application.

As we need the technical ID of the property we have to select this property in the property editor of the related iView. Using the small triangle icon on the left side we could display the *meta attributes* of the property as shown in the next picture.



Tray Type: PLAIN

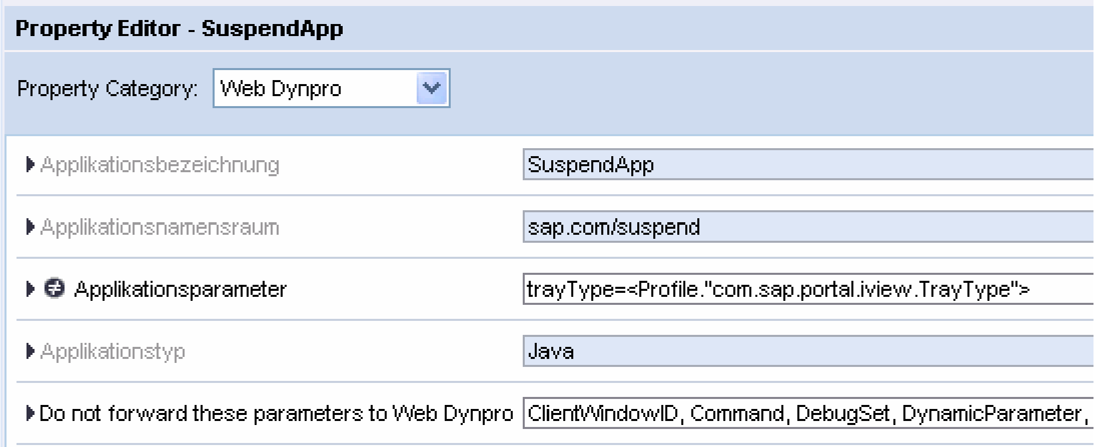
Property ID: com.sap.portal.iview.TrayType

Inheritance: Can be Edited in Target Objects

End-User Personalization: Hidden

Property Description: Type of tray displayed behind and around the iView/page

The meta attribute `Property ID` describes the technical name. Because `com.sap.portal.iview.TrayType` contains special characters (`.`), you must specify the URL template variable using the following syntax: `<Profile."com.sap.portal.iview.TrayType">`. By doing this, you can ensure that the URL template will be processed correctly. The following picture shows the definition of the variable using the `Application Parameters` iView property.



Property Editor - SuspendApp

Property Category: Web Dynpro

Applikationsbezeichnung	SuspendApp
Applikationsnamensraum	sap.com/suspend
Applikationsparameter	trayType=<Profile."com.sap.portal.iview.TrayType">
Applikationstyp	Java
Do not forward these parameters to Web Dynpro	ClientWindowID, Command, DebugSet, DynamicParameter,

You can now access the forwarded `trayType` parameter using the `WDProtocolAdapeter` class as shown in the following code example:

```
String trayType =
        WDProtocolAdapter().
        getProtocolAdapter().
        getRequestObject().
        getParameter(“trayType”);
```

Using NW04s Web Dynpro iViews

One of the main differences between NW04 Web Dynpro iViews and NW04s Web Dynpro iViews is the way how the Web Dynpro application is executed. For NW04 iViews this is done by computing a URL pointing to the Web Dynpro application. This URL is the only relationship between the SAP NetWeaver portal and the Web Dynpro runtime. All needed information must be passed therefore together with this URL.

For NW04s Web Dynpro iViews the Web Dynpro page builder is responsible to launch the Web Dynpro application. As the Web Dynpro page builder and the launched application must run on the same machine this is done not using an HTTP request but through specific server-side APIs between the Web Dynpro page builder and the Web Dynpro runtime. This means on the other side that a Web Dynpro application running as an NW04s Web Dynpro iView could assume beforehand that a SAP NetWeaver portal is installed and running on the same machine as the Web Dynpro application itself.

We therefore could directly access the PCD within a Web Dynpro application as shown in the following code example:

```
try {
  Hashtable env = new Hashtable();
  env.put(Constants.REQUESTED_ASPECT,
        PcmConstants.ASPECT_SEMANTICS);
  env.put(Context.SECURITY_PRINCIPAL,
        WDClientUser.getCurrentUser().getSAPUser());

  InitialContext iCtx = new InitialContext(env);
  IAttributeSet roleAttributeSet = (IAttributeSet) iCtx.lookup(iViewPath);

  String trayType
        = attributeSet.getAttribute (“com.sap.porta.iview.TrayType”);

 } catch (Exception e) {
       // ....
 }
```

Technically spoken we access the PCD through a standard JNDI lookup. If you would like to access the iView properties including also the end-user personalization you have to add an additional parameter to the used JNDI environment as shown in the next example:

```
Hashtable env = new Hashtable();
env.put(Constants.REQUESTED_ASPECT,
  PcmConstants.ASPECT_SEMANTICS);
env.put(Context.SECURITY_PRINCIPAL,
  WDClientUser.getCurrentUser().getSAPUser());
env.put(IPcdContext.PCD_PERSONALIZATION_PRINCIPAL,
  WDClientUser.getCurrentUser().getSAPUser());
```

Although this approach is valid and a working solution you have to keep in mind that you are bound to the fact that the Web Dynpro application is running on the same SAP NetWeaver installation as the SAP NetWeaver Portal itself. You must consider this restriction before going into this direction.

To get more details about this proposal please have a deeper look to the Maximizing Web Dynpro for Java book, where we give much more details and we even describe a work-around to define application specific iView properties for a Web Dynpro iView.

Outlook

We would like to give also a small outlook to the next SAP NetWeaver release. We will introduce with this release the capability to expose standard Web Dynpro application properties to the portal. These application properties could be then customized as any other iView property. Changes are automatically passed to the Web Dynpro runtime as soon as a Web Dynpro application is launched. Within the Web Dynpro application you have access to the changed values using the standard APIs to access application properties.

Implicit Web Dynpro personalization

General idea

Besides the possibility to define explicitly specific properties which could be maintained later on for a specific user or a group of users, there are scenarios needing another mechanism. Image the personalization of UI elements for example: hiding an input field for a not needed field, disabling a button for a group of users, which does not have the needed permissions to execute the action associated to the button, or changing the order of certain table columns are only some few examples for this. Theoretically it would be of course possible to define all needed parameters for this explicitly but for obvious reasons this it not a usable solution.

To provide a better solution for the described scenarios we introduced the *implicit Web Dynpro personalization*, which allows you to manipulate a lot of aspects of almost all available UI elements - without defining any explicit properties and without any application specific coding. As we will describe later in more detail the implicit personalization could be used both for end-user personalization and personalization for a group of users.

Architecture

The basis idea of the implicit personalization for Web Dynpro application is the following:

- The Web Dynpro runtime defines for each UI element the list of properties which could be personalized by a single user and for a group of users.
- The Web Dynpro runtime offers generic personalization dialogs to maintain the different properties of an UI element. End-user personalization and personalization for a group of users is handled by different personalization dialogs, which are optimized for the needed usecases. Both kinds of personalization dialogs could be opened pressing the **CNTRL** key together with the right mouse button on the UI element, which should be personalized.
- As soon as a Web Dynpro application is launched as an NW04s Web Dynpro iView within the SAP NetWeaver portal, the stored personalization data (both for end user personalization and personalization for a group of users) is passed to the application instance and the Web Dynpro runtime ensures that the displayed UI is adjusted accordingly.
- The created personalization data is stored in the *Portal Content Directory (PCD)*. The data is handled separately for each NW04s Web Dynpro iView instance. The delta link mechanism of the PCD is working also for the implicit Web Dynpro personalization.
- From a technical point of view this data is stored as a list of iView properties. In contrast with the explicit iView personalization there is no need to explicitly access these properties. Only in error scenarios it could make sense to check single properties (we will describe this later in this document).

Prerequisites

The implicit Web Dynpro personalization works only when running a Web Dynpro application as an NW04s Web Dynpro iView. This dependency is based on the fact that the Web Dynpro page builder is responsible to provide the stored personalization data to the Web Dynpro application during startup. The Web Dynpro page builder is also responsible to store new personalization data created using the implicit Web Dynpro personalization. This data is passed back from the Web Dynpro application to the Web Dynpro page builder as soon as on of the personalization dialogs are used to personalize the application either or an end-user not a group of users.

Single user vs. group of users

As described above there are in general two different flavors of personalization: the end-user personalization and the personalization for a group of users. The implicit personalization of Web Dynpro applications handles these two scenarios using two different modes: the *end user mode* and the *administrator mode*.

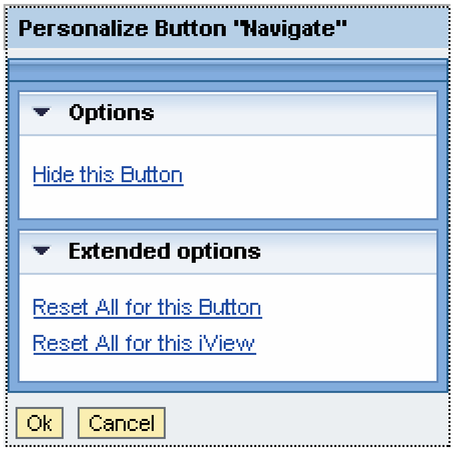
We will describe the capabilities and the differences of these two modes in the following two sub chapter.

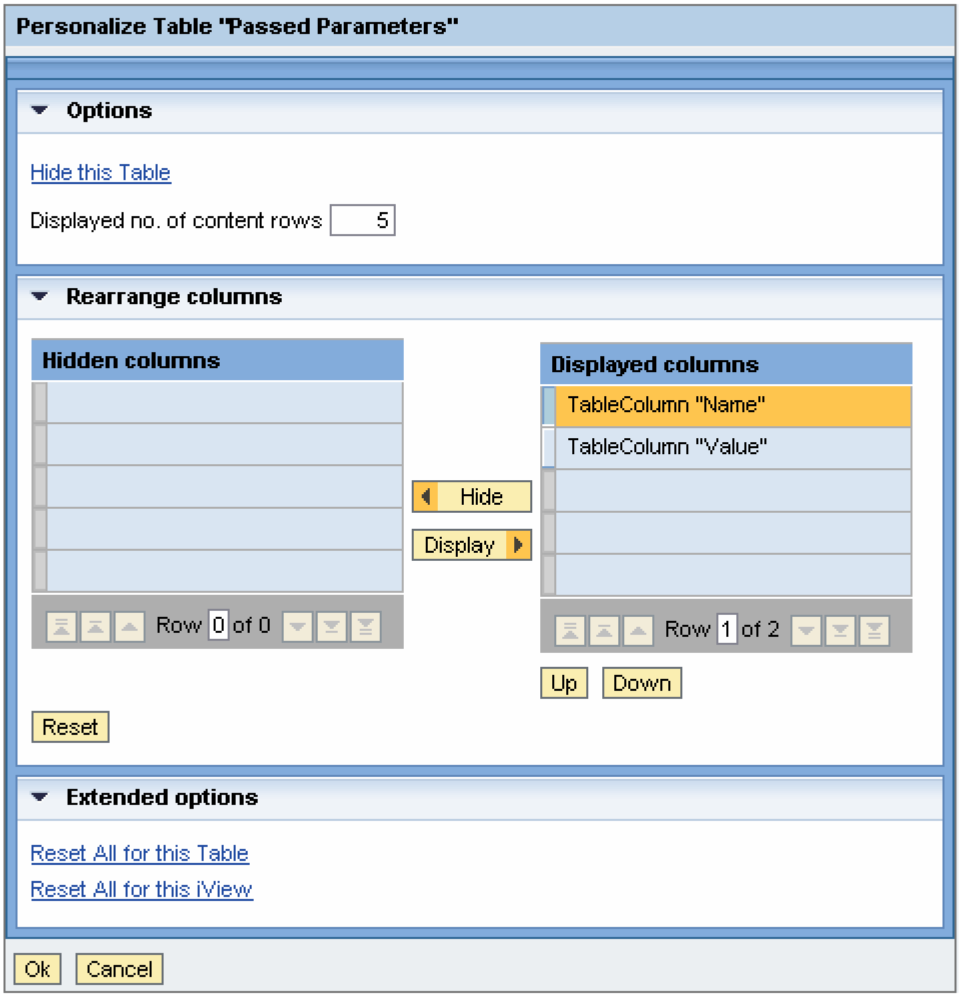
End-user mode

Starting the implicit personalization in the end-user mode is allowed for all users (surprise, surprise).

All changes are stored only for the current user. If other users execute the same NW04s Web Dynpro iView the changes done in the end-user mode are not shared between different users. If an NW04s Web Dynpro iView is used in different pages or roles these iView instances are typically connected using the delta link mechanism of the PCD. As the personalization data created by the implicit personalization is stored in the end as standard iView properties (as explained above) the same delta link mechanism is available also for the implicit personalization.

The following pictures shows typical personalization dialogs for the end-user mode. The first one shows the personalization dialog for a button, the second one the dialog a table.





Personalize Table "Passed Parameters"

Options

Displayed no. of content rows 5

Rearrange columns

Hidden columns

Row 0 of 0

Displayed columns

TableColumn "Name"

TableColumn "Value"

Row 1 of 2

Extended options

The shown personalization dialogs demonstrates very well the general capabilities of the implicit personalization running in the end-user mode as described in the following:

- UI elements could be hidden by clicking the appropriate **Hide …** link in the personalization dialog.
- Hidden UI elements could be made visible again using the appropriate **Show …** links under the **Hidden controls** tray.
- For tables you can define the number of visible rows. Besides that also the list of visible columns and their order could be changed.
- Under the **Extended options** tray you could reset all end-user related changes done for the selected UI element or even done for the complete iView instance.

Administrator mode

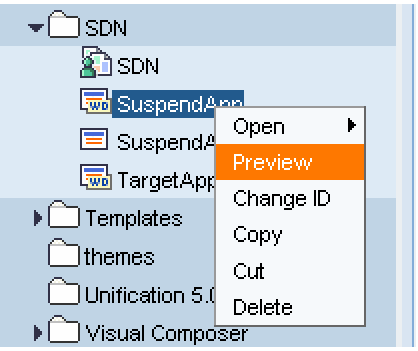
To run the implicit personalization in the administrator mode you need *content administration* permissions for the SAP NetWeaver Portal.

The handling of the created personalization data is the same as in the end-user mode. The data is stored in the end as standard iView properties and the delta link mechanism of the PCD would work in case that the implicit personalization is used for several iViews (in different page, roles) connected by delta links.

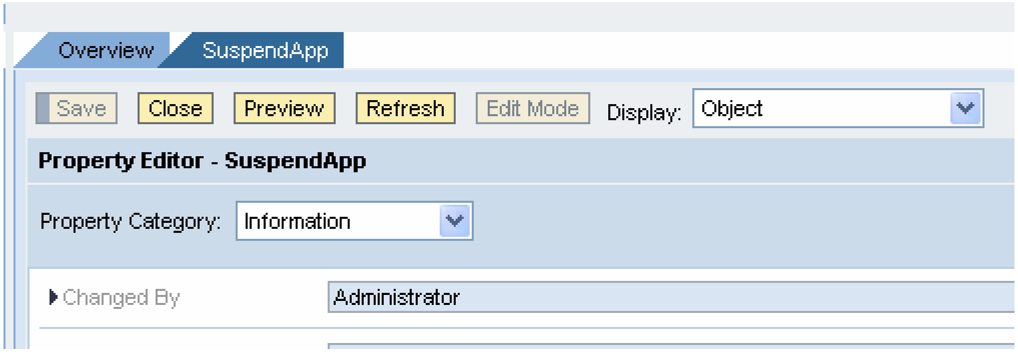
To run the implicit personalization in the administrator mode you have to run the *preview* of the iView you want to customize. To run the preview you have to navigate to the iView using the portal content catalog available under **Content Administration -> Portal Content**.

If you want to personalize the base iView you could run the preview of this iView in two different ways:

- You could use the **Preview** entry of the iView context menu as shown on the next picture.



- You could open the iView editor using the **Open -> Object** context menu entry of the iView and press the **Preview** button at the top of the iView editor



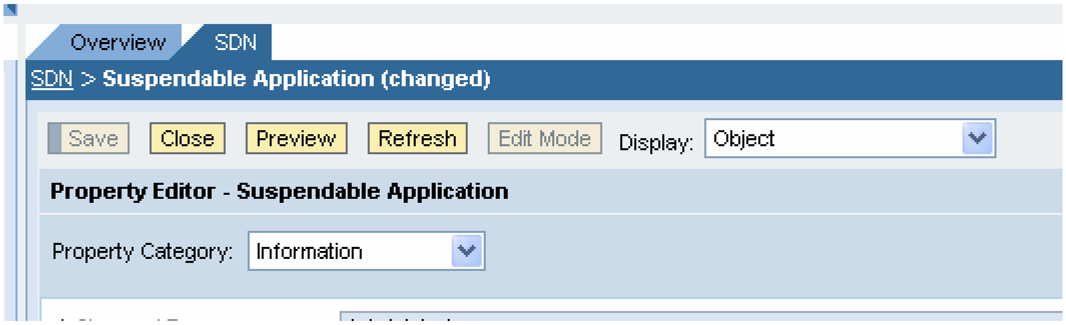
Display: Object

Property Editor - SuspendApp

Property Category: Information

Changed By: Administrator

If you want to customize an iView within a certain page or role you have to make sure that you do *not* open the base iView but the iView (instance) inside the page or role. The preview of an iView inside a page or role could be launched using the **Preview** button on top of the iView editor as shown in the next screenshot.



SDN > Suspendable Application (changed)

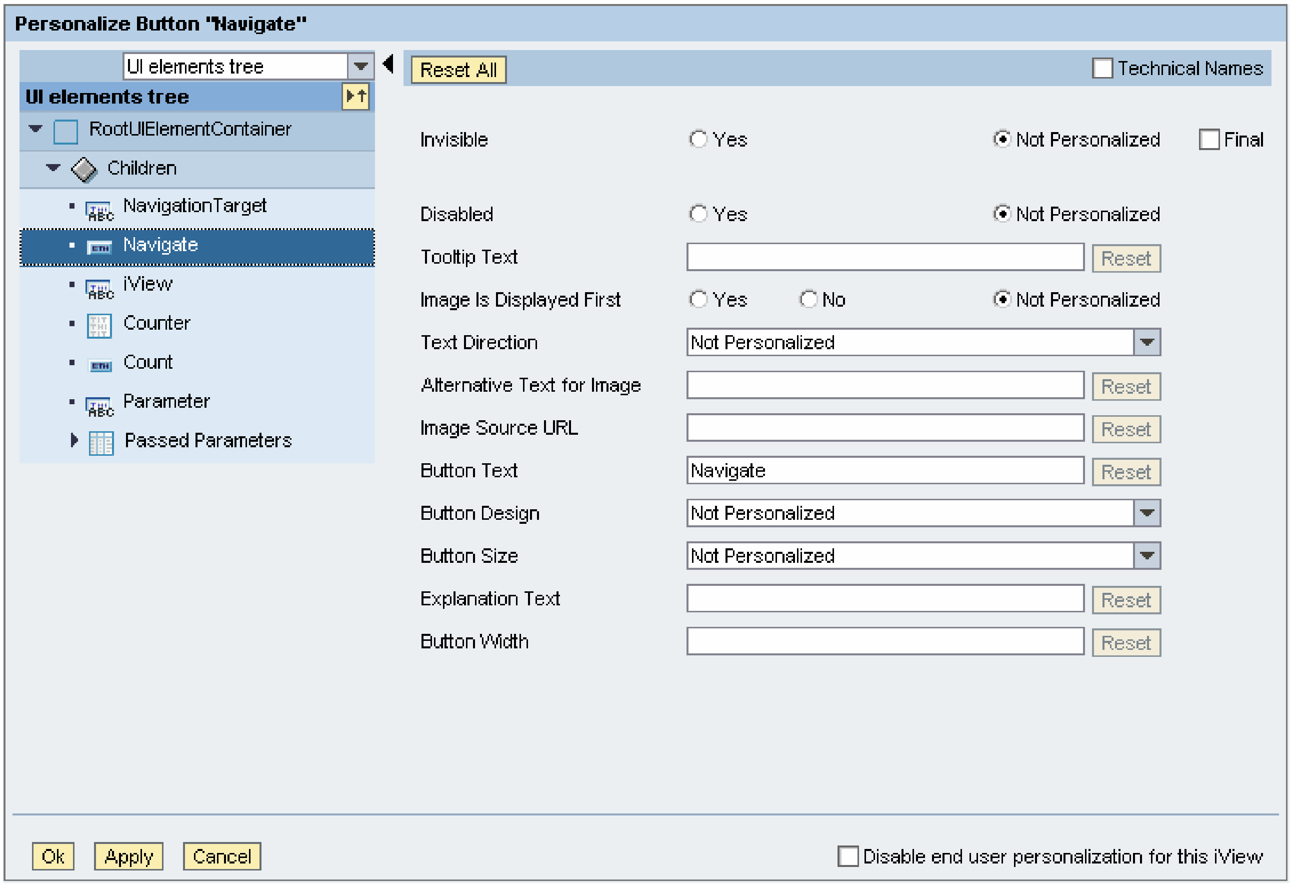
Display: Object

Property Editor - Suspendable Application

Property Category: Information

Please consider the displayed iView path `SDN -> Suspendable Application` in the title bar of the iView editor which visualizes that the iView is opened within the `SDN` role.

Running the implicit personalization in the administrator mode provides much more capabilities as the end-user mode. The personalization dialog opened for a certain UI element always looks like the same as shown in the next picture.



Personalize Button "Navigate"

□ Technical Names

Invisible ○ Yes ● Not Personalized □ Final

Disabled ○ Yes ● Not Personalized

Tooltip Text [ ] Reset

Image Is Displayed First ○ Yes ○ No ● Not Personalized

Text Direction: Not Personalized

Alternative Text for Image [ ] Reset

Image Source URL [ ] Reset

Button Text: Navigate Reset

Button Design: Not Personalized

Button Size: Not Personalized

Explanation Text [ ] Reset

Button Width [ ] Reset

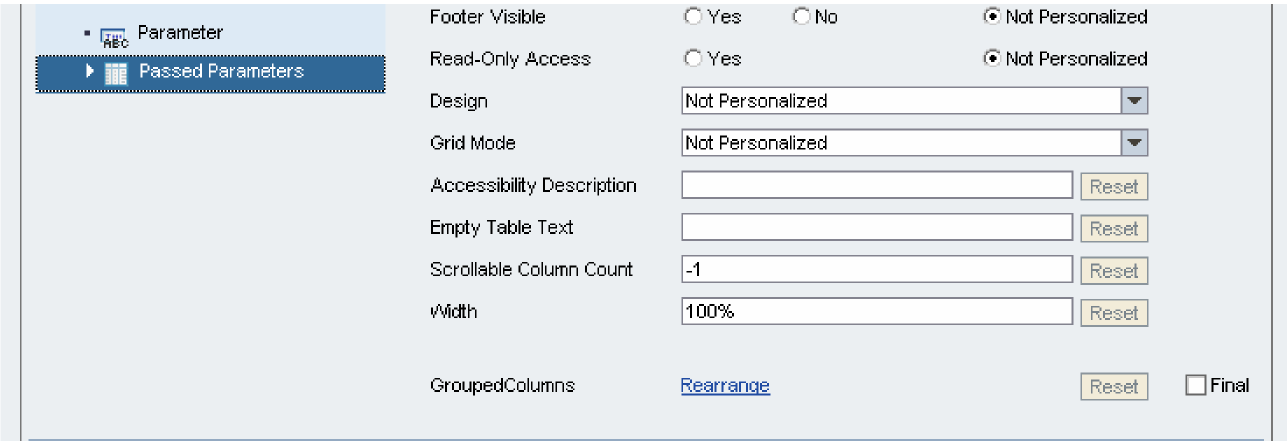
□ Disable end user personalization for this iView

The main difference to the end-user mode is the possibility to customize not only the selected UI element but the whole UI element tree of the current screen. Using the navigation panel on the left side you can select any UI element. The shown properties on the right side are adjusted accordingly as soon as a UI element is selected.

In spite of the end-user mode which allows only changes of the visibility of an UI element there are much more properties available in the administrator mode. Besides the visibility you could change almost every property of an UI element.

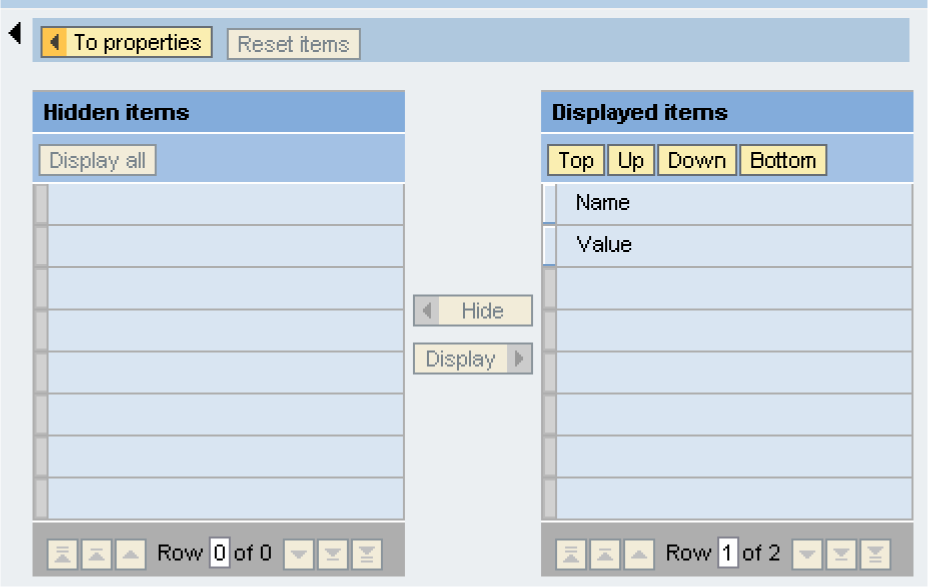
For all properties which could be also changed by an end-user you can define the changes as final using the **Final** check boxes on the right side of a property. A final change does mean that this property could not be changed later on using the end-user mode of the implicit personalization.

As soon as you select a table element you could also maintain the list of visible columns and their order by pressing the **Rearrange** link on the right side as shown in the next picture.



Footer Visible	○ Yes	○ No	● Not Personalized
Read-Only Access	○ Yes		● Not Personalized
Design	Not Personalized		
Grid Mode	Not Personalized		
Accessibility Description			Reset
Empty Table Text			Reset
Scrollable Column Count	-1		Reset
Width	100%		Reset
GroupedColumns	Rearrange		Reset □ Final

The right side of the personalization dialog is replaced in this case with a screen providing you the possibility to hide or show certain columns and to define their order.



Hidden items

Row 0 of 0

Displayed items

Name
Value

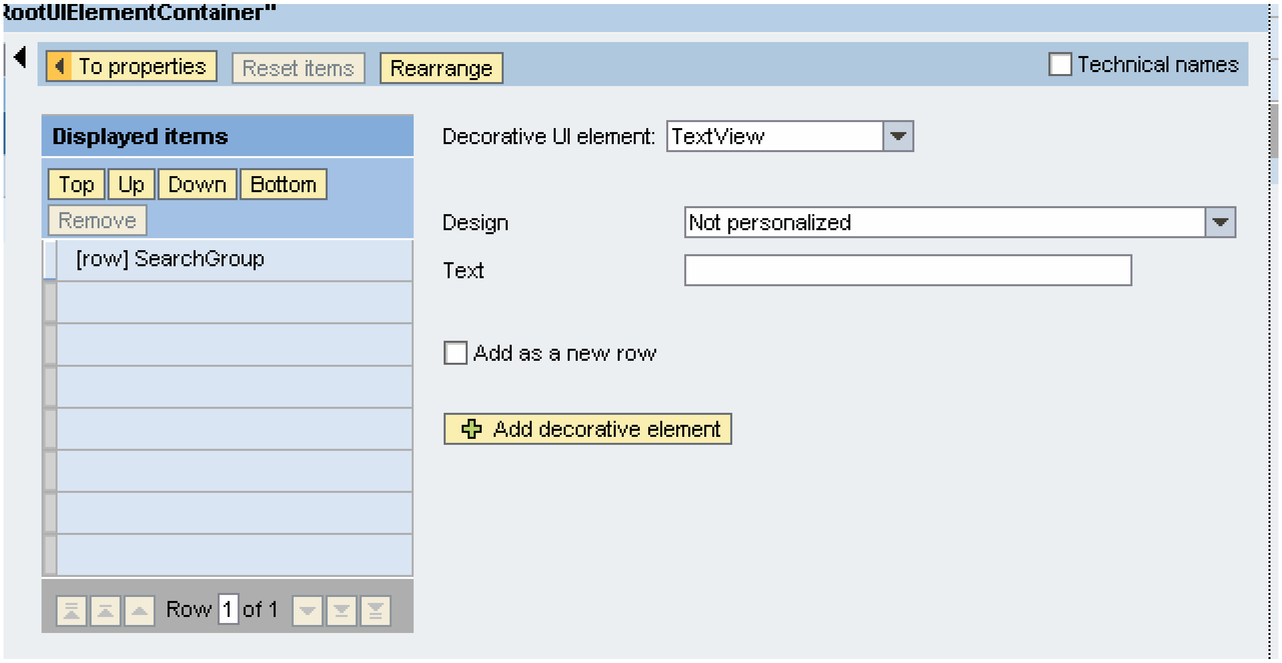
Row 1 of 2

To navigate back to the properties view you could use the **To properties** button or you could select any simple UI element using the navigation panel on the left side.

As soon as you select a container UI element (like an `IWDTransparentContainer` UI element or an `IWDGroup` UI element) you can also change the visibility and order of the children of the container (similar to the columns of a table).

Besides that you can also add decorative elements to a container like an additional `IWDTextView` UI element to add more infos on the screen or a `IWDLinkToURL` UI element to provide a additional link pointing to more infos about an application.

The next picture shows the personalization dialog when adding decorative elements to a container UI element. To switch to this mode you have to press the **Decorate** link shown as soon as a container UI element is selected in the navigation panel.



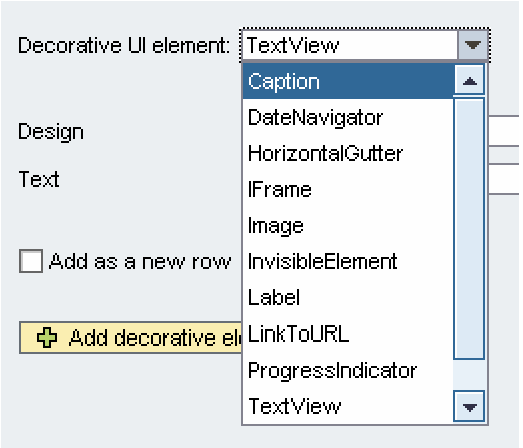
Decorative UI element: TextView

Design: Not personalized

Text:

After selecting the decorative UI element you can define the needed design and text values. Pressing the **Add decorative elements** button adds this element to the screen.

You can choose the decorative element from a long list of UI elements as shown in the next picture.



Decorative UI element: TextView

To disable the implicit personalization in the end-user mode for an iView in general, you could use the **Disable end user personalization for this iView** check box.

Field-extensions

Besides the described functionality of the administrator mode of the implicit Web Dynpro personalization there is one additional very powerful feature: the possibility to define field extensions (especially together with the adaptive RFC model to access backend data within a SAP system). Please check the detailed documentation about this special feature of the implicit Web Dynpro personalization available also under sdn.sap.com.

Transport of personalization data

Keeping in mind that the personalization data created using the implicit Web Dynpro personalization is stored as a set of iView properties helps to understand how personalization data could be transported. Using the standard import / export mechanism for portal content it is also possible to transport “personalized” Web Dynpro iViews. As for the explicit iView personalization only personalization data created for a group of users could be transported to another system. End-user personalization is always defined locally and it could not be transported to another system.

To import or export portal content you have to use the standard PCD transportation tools available under **System Administration -> Transport –> Transport Packages**.

Handling errors

All data which is created using the implicit Web Dynpro personalization is stored in the end as standard iView properties as described above. The transport of this data is done completely transparent for the Web Dynpro application. The Web Dynpro page builder ensures that the stored data is forwarded to the Web Dynpro application as soon as the application is started as NW04s Web Dynpro iView. The Web Dynpro page builder is also responsible to store new personalization data.

But how to handle scenarios, where this does not work in the expected way? How do you get more information in cases that the personalized UI is not the expected one? Is there a way to find out, whether the data is stored correctly or which data is stored in general?

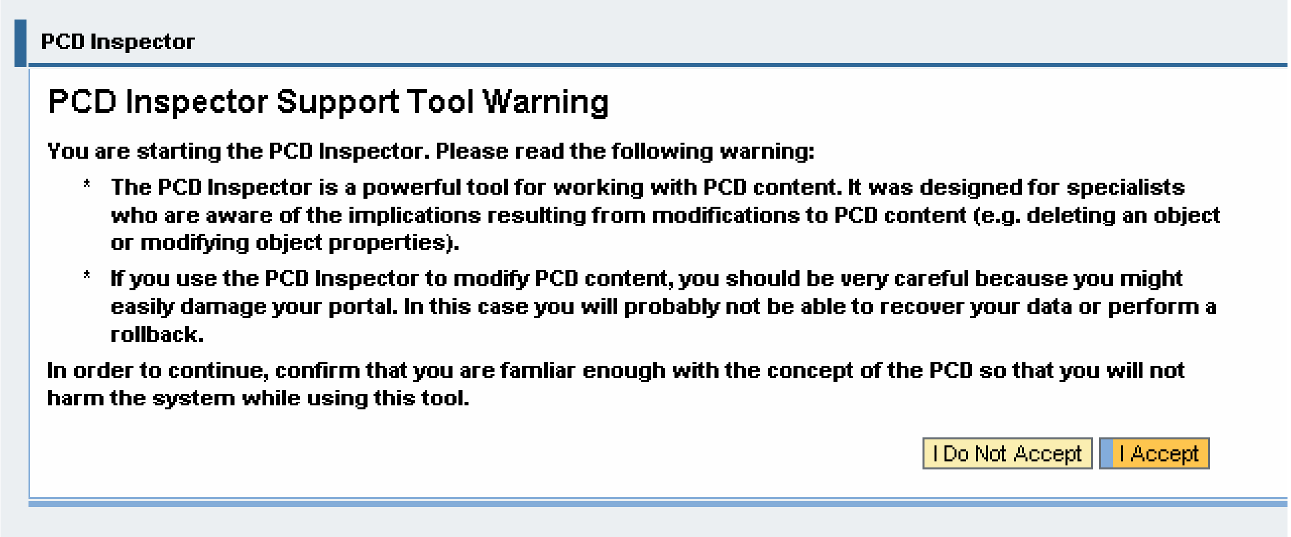
Two answers for these questions are given in the next two sub chapters.

The PCD Inspector

Although the personalization data created by the implicit Web Dynpro personalization is stored as standard iView properties, these iView properties are not visible using the iView editor. The iView editor shows only iView properties which could be changed by an administrator or even an end-user. The personalization data of the implicit Web Dynpro personalization is not usable for this.

Nevertheless there is an option to check the stored personalization data using the *PCD Inspector*, which could be found under **System Administration -> Support -> Portal Content Directly -> PCD Inspector**.

As the PCD Inspector is a very generic tool which give you a lot of capabilities to change the PCD data you are warned when you start it the very first time as shown in the next picture.


PCD Inspector

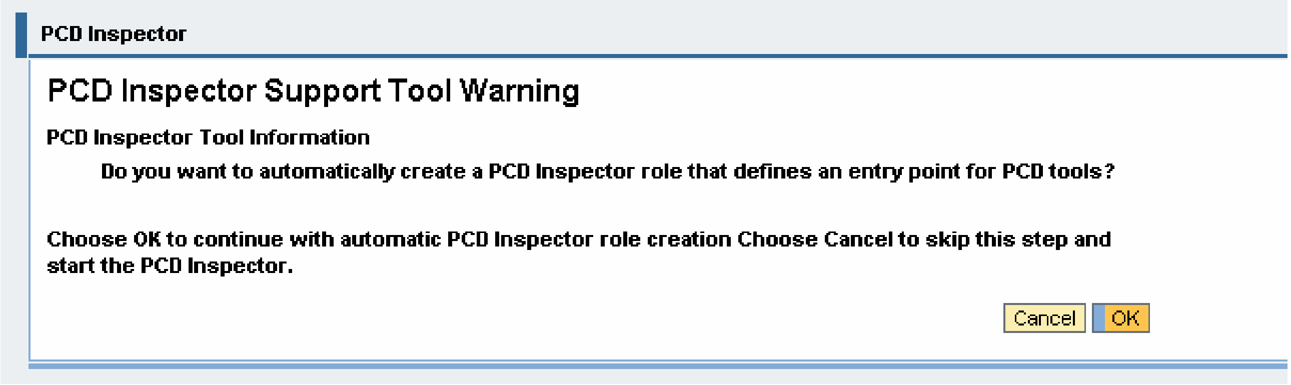
PCD Inspector Support Tool Warning

You are starting the PCD Inspector. Please read the following warning:

* The PCD Inspector is a powerful tool for working with PCD content. It was designed for specialists who are aware of the implications resulting from modifications to PCD content (e.g. deleting an object or modifying object properties).
* If you use the PCD Inspector to modify PCD content, you should be very careful because you might easily damage your portal. In this case you will probably not be able to recover your data or perform a rollback.

In order to continue, confirm that you are famliar enough with the concept of the PCD so that you will not harm the system while using this tool.

After pressing the **I Accept** button you are asked whether you want to create a certain PCD Inspector role, which allows you an easy access to it later one. We recommend to do this and to press the **OK** button.



PCD Inspector

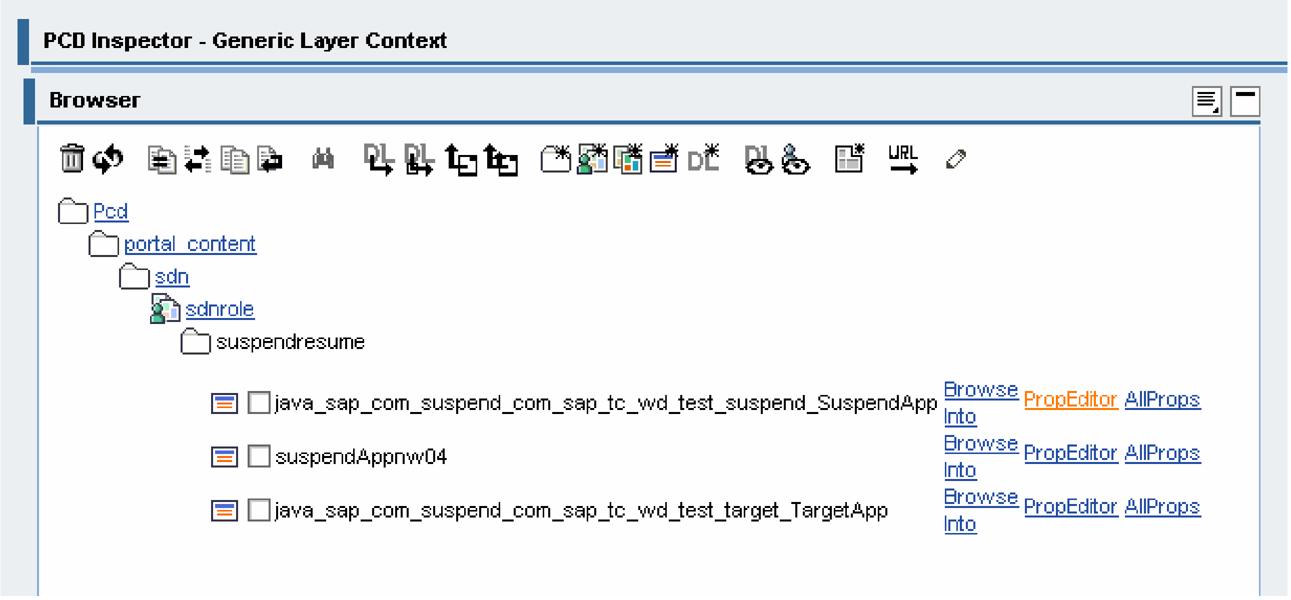
PCD Inspector Support Tool Warning

PCD Inspector Tool Information

Do you want to automatically create a PCD Inspector role that defines an entry point for PCD tools?

Choose OK to continue with automatic PCD Inspector role creation Choose Cancel to skip this step and start the PCD Inspector.

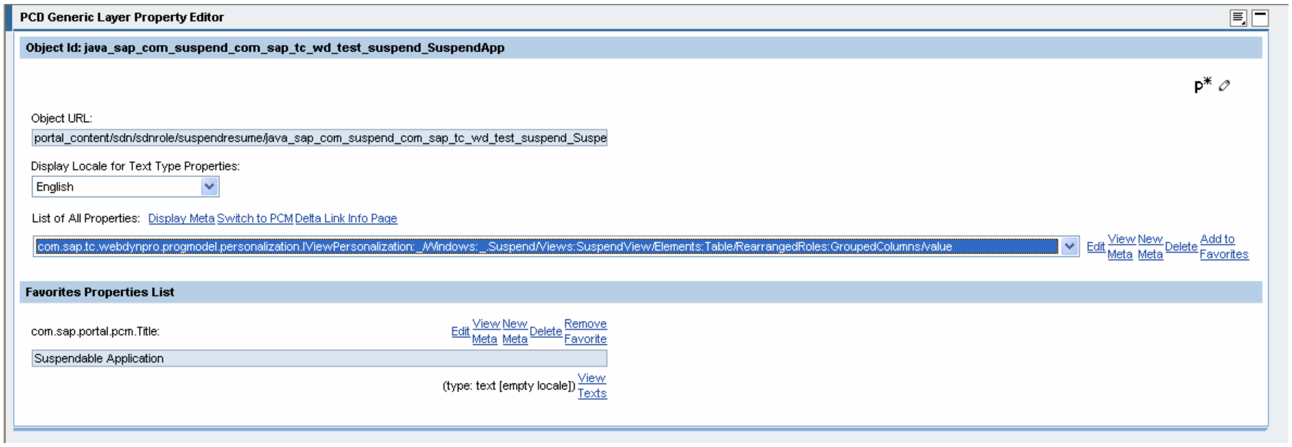
You can now browse the complete data stored in the PCD. The portal content is located under the `portal_content` folder. To drill down one level you have to choose the **Browse Into** link.



□ java_sap_com_suspend_com_sap_tc_wd_test_suspend_SuspendApp	Browse Into	PropEditor	AllProps
□ suspendAppnw04	Browse Into	PropEditor	AllProps
□ java_sap_com_suspend_com_sap_tc_wd_test_target_TargetApp	Browse Into	PropEditor	AllProps

To display the properties for a certain object you have to open the property editor using the **PropEditor** link.

The next picture shows the property editor displaying an attribute created by the implicit Web Dynpro personalization.



PCD Generic Layer Property Editor

Object Id: java_sap_com_suspend_com_sap_tc_wd_test_suspend_SuspendApp

Object URL: portal_content/sdn/sdnrole/suspendresume/java_sap_com_suspend_com_sap_tc_wd_test_suspend_Suspe

Display Locale for Text Type Properties: English

List of All Properties: com.sap.tc.webdynpro.progmodel.personalization.IViewPersonalization:_/Windows:_Suspend/Views:SuspendView/Elements:Table/RearrangedRoles:GroupedColumns/value

Favorites Properties List

com.sap.portal.pcm.Title: Suspendable Application

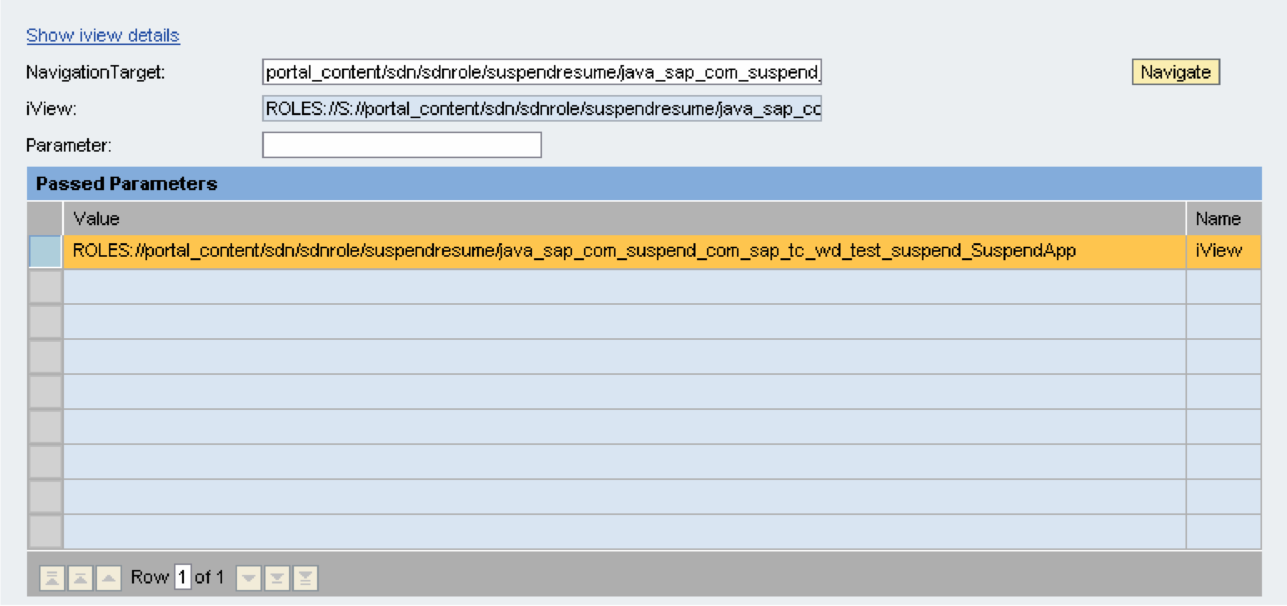
(type: text [empty locale])

The Web Dynpro page builder debug screen

The Web Dynpro page builder debug screen provides you additional information about the communication between the Web Dynpro page builder and the embedded Web Dynpro application (running as NW04s Web Dynpro iView).

To turn on this debug screen you have to use the Web Dynpro Debug Configurator as described in the related document.

The following screenshot show a simple Web Dynpro application running as NW04s Web Dynpro iView. As soon as the Web Dynpro page builder debug mode is turned on there is a additional **Show / Hide iview details** link , which allows you to show and hide additional data of the iView.



NavigationTarget: portal_content/sdn/sdnrole/suspendresume/java_sap_com_suspend

iView: ROLES://S://portal_content/sdn/sdnrole/suspendresume/java_sap_cc

Parameter:

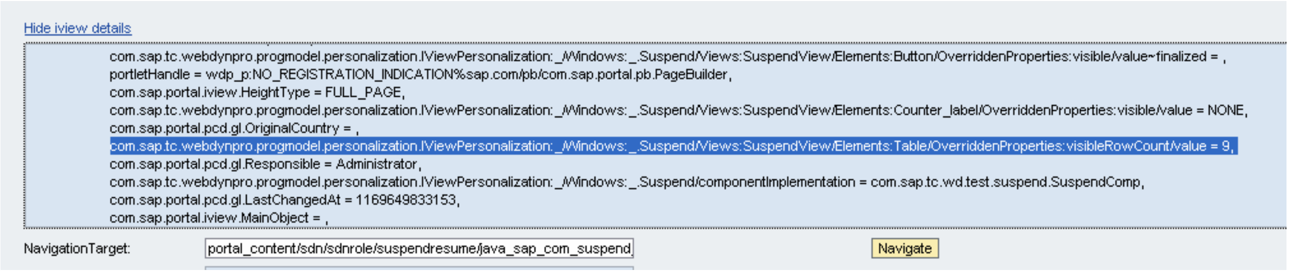
Passed Parameters

Value	Name
ROLES://portal_content/sdn/sdnrole/suspendresume/java_sap_com_suspend_com_sap_tc_wd_test_suspend_SuspendApp	iView

Row 1 of 1

Regarding the implicit Web Dynpro personalization the displayed personalization data is important as these two maps shows the data, which is forwarded from the Web Dynpro page builder to the Web Dynpro runtime.

The next picture shows for example the attribute defining the personalization of the number of visible rows.



```
com.sap.tc.webdynpro.progmodel.personalization.IViewPersonalization:_/Windows:_.Suspend/Views:SuspendView/Elements:Button/OverriddenProperties:visible/value~finalized = ,
portletHandle = wdp_p:NO_REGISTRATION_INDICATION%sap.com/pb/com.sap.portal.pb.PageBuilder,
com.sap.portal.iview.HeightType = FULL_PAGE,
com.sap.tc.webdynpro.progmodel.personalization.IViewPersonalization:_/Windows:_.Suspend/Views:SuspendView/Elements:Counter_label/OverriddenProperties:visible/value = NONE,
com.sap.portal.pcd.gl.OriginalCountry = ,
com.sap.tc.webdynpro.progmodel.personalization.IViewPersonalization:_/Windows:_.Suspend/Views:SuspendView/Elements:Table/OverriddenProperties:visibleRowCount/value = 9,
com.sap.portal.pcd.gl.Responsible = Administrator,
com.sap.tc.webdynpro.progmodel.personalization.IViewPersonalization:_/Windows:_.Suspend/componentImplementation = com.sap.tc.wd.test.suspend.SuspendComp,
com.sap.portal.pcd.gl.LastChangedAt = 1169649833153,
com.sap.portal.iview.MainObject = ,
```

NavigationTarget: portal_content/sdn/sdnrole/suspendresume/java_sap_com_suspend

Copyright

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.