Web Dynpro for ABAP: Tutorial 5 – Component and Application Configuration

SAP NetWeaver 2004s

Copyright

Icons in Body Text

Icon	Meaning
⚠	Caution
💬	Example
💡	Note
🧭	Recommendation
SYN	Syntax

Additional icons are used in SAP Library documentation to help you identify different types of information at a glance. For more information, see *Help on Help* → *General Information Classes and Information Classes for Business Information Warehouse* on the first page of any version of *SAP Library*.

Typographic Conventions

Type Style	Description
Example text	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation.
`Example text`	Emphasized words or phrases in body text, graphic titles, and table titles.
EXAMPLE TEXT	Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.
`Example text`	Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
`Example text`	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
`<Example text>`	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
`EXAMPLE TEXT`	Keys on the keyboard, for example, `F2` or `ENTER`.

Web Dynpro for ABAP: Tutorial 5 – Component and Application Configuration

Preface

In Tutorial 3 you enhanced the Web Dynpro flight sample with a navigation to a separate view including a message for the case where no flights are found for a given search query. But some of your customers might prefer the former solution where the empty result table is displayed instead of jumping to a new view.

One possibility would be to have two slightly different Web Dynpro components. But it would obviously be better to have just one Web Dynpro component which can be configured somehow to behave in one or the other way. You can imagine that a real life application can have several such switches or parameters to steer the behavior of such a flexible application.

Therefore, Web Dynpro provides several techniques, such as configuration, personalization, or customization, in order to modify the UI or the navigation path of a delivered Web Dynpro application or component at the customer's site.

This tutorial shows how to implement such a configuration for this simple example with two different possibilities to handle the case of no found flights.

Configuration vs. Personalization and Customization

When using standard software in different companies, the requirements of the user interface and application behaviour might differ enormously. Applications that are created with the help of Web Dynpro for ABAP can be adjusted in different ways and by different target groups. In this context, we distinguish between two areas:

- Configuration

  and

- Personalization and Customization (Cross-Client Adjustments)

Configuration

Configuration modifies the behaviour of a Web Dynpro component and application for all users of the application in the same way. A Web Dynpro component can have special parameters which steer the behavior of the component. A set of parameter values which define how the component should act is called a configuration, or configuration data record. Think, for example, of a component with two different configurations (set of values) one for an experienced user (more options are explained) and one for the casual user (less options but more explaining texts) of the application.

The configuration of business applications takes place in two subsequent steps:

1. First, configuration data records are created for each individual Web Dynpro component (= *Component Configuration*) - remember: a Web Dynpro application can consist of different Web Dynpro components. Such configuration data records are absolutely necessary when using generic components, such as the ALV or Pattern component. Configuration data is created and edited mainly by developers.

   There are two flavours of component configuration: implicit and explicit configuration. One type (implicit) is directly available as predefined parameters of the UI elements where the UI elements are displayed by the WD framework according to the parameter settings automatically. With the other type (explicit) the influence on the components behaviour has to be explicitely implemented by the programmer and just the parameter values are stored by the framework.

   a. *Implicit Configuration* modifies the properties of an UI element (such as Visibility) which was set at design time by the developer to a special value but can be overwritten using configuration tools. There is no additional programming effort required by the component developer for implicit configuration. Many (not all) of the attributes of UI elements can be overwritten with the help of configuration data.

   b. When using *Explicit Configuration* it is possible to influence the appearance and behavior of a Web Dynpro component. These influences can be implemented through additional attributes in the context of a special custom controller - the Configuration Controller - where you can define attributes in the context of this controller. The storage of the attributes in the context of the configuration controller ensures that their values can be set or changed later on with the help of the *Component Configurator*.
   Furthermore, the processing of attributes must be explicitly programmed by the application developer in a controller method of the component.

2. The following step is the *Application Configuration*.
   Each of the used Web Dynpro components is required in a particular configuration. The application configuration defines which component with which configuration is required for the application. This step, too, is carried out primarily by application developers using the *Application Configurator*.

Hence, configuration of Web Dynpro components and applications almost always takes place at design time (the only exception here is the configuration of a bound ALV component).

Personalization and Customization

In contrast to configuration, personalization is a function that is available to the user of an application and provides him or her with the option of adjusting the application to suit his or her own personal requirements or preferences. The framework for variation options within personalization is less far-reaching than that in configuration; personal settings in the UI must never limit the running ability of an application. Personalization of an application is performed directly by a user from within the current application.

Furthermore, it is possible to maintain personalization settings in a uniform manner for larger user groups (cross-client adjustment - also called *Customization*). A system administrator can process personalization settings on the basis of his or her extended authorization provided the respective application runs in so-called configuration mode.

Personalization and customization are always executed at runtime of an application.

Development Objectives

This exercise demonstrates the usage of explicit component configuration and application configuration.

Our Web Dynpro example of Tutorial 3 displays a separate view with a message if no flight is found between the selected cities. Some customers might prefer to simply display the normal result view with an empty table in such a case. Therefore the Web Dynpro component should be enhanced with the possibility to configure this behavior.

At the end of this exercise you will be able to

- Understand the concept of Web Dynpro component and application configuration
- Use the Component Configurator to enhance a Web Dynpro component with configuration capabilities
- Use the Application Configurator to create Web Dynpro configurations which steer the behavior of a Web Dynpro application

> ⚠ Procedures, which already were explained in detail in Tutorial 1, for example,
>
> - how to activate a Web Dynpro component, or
> - how to execute a Web Dynpro application,
>
> are not explained in detail in this tutorial. Therefore, we strongly recommend, that you first thoroughly work through Tutorial 1 before you start this Tutorial.

Procedure

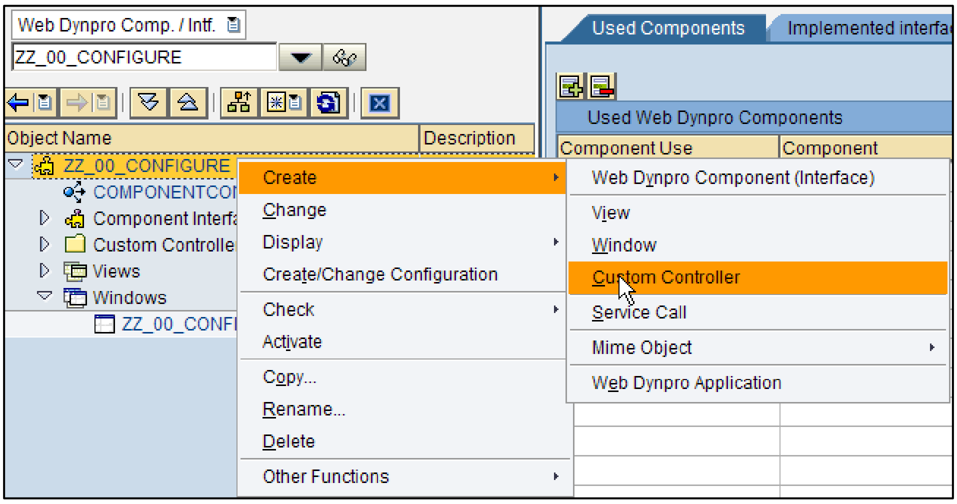
1. Copy the provided component ZZ_00_BAPINAV to ZZ_00_CONFIGURE.
2. Create a new custom controller *NAVIGATION_CONFIG* and set it as Config. Controller.
3. Add a node NAV_PARAMS and an attribute NAVTO_NOFLIGHTSVIEW of type WDY_BOOLEAN to the context of the config controller.
4. Define the controller usage of NAVIGATION_CONFIG to the view context of FLIGHTLISTVIEW and map the node NAV_PARAMS to the view context.
5. Enhance the method ONACTIONGET_FLIGHTS of the view FLIGHTLISTVIEW, so that the navigation to the NOFLIGHTSVIEW is only triggered if the attribute NAV_TO_NOFLIGHTSVIEW is true (= ABAP_TRUE).
6. Save and activate the component. Create and test the corresponding application. If no flight is found an empty table will be shown because the attribute NAV_TO_NOFLIGHTSVIEW is false by default.
7. Create a new Web Dynpro application configuration ZAC_SHOW_NOFLIGHTS for the Web Dynpro application.
8. In the application configuration editor create also a corresponding Web Dynpro component configuration ZCC_SHOW_NOFLIGHTS and check the default value of the attribute NAV_TO_NOFLIGHTSVIEW.
9. Test this application configuration by selecting two exotic airports like ALICE SPRINGS and BASEL.
10. You can also create a Web Dynpro application configuration and Web Dynpro component configuration with the flag unchecked.

Create the Web Dynpro Component ZZ_00_CONFIGURE

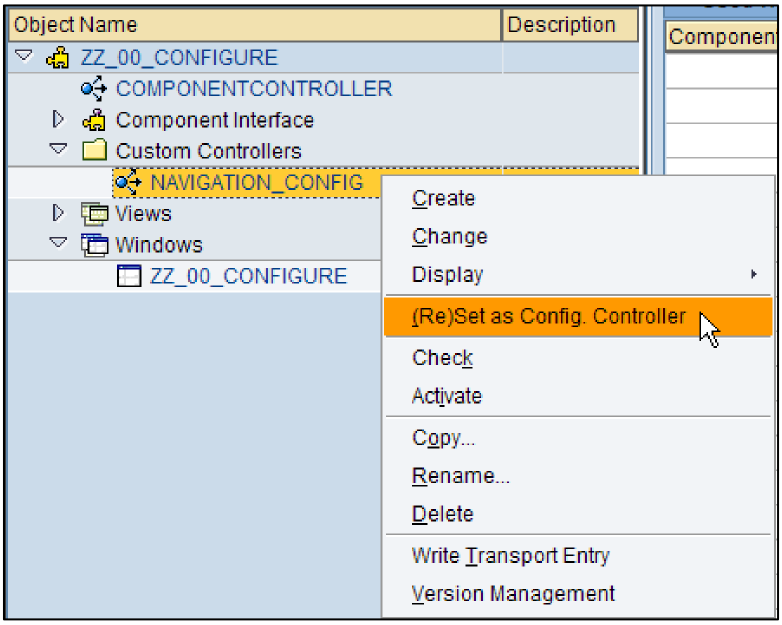
1. Copy the the result of Tutorial 3 (which should be Web Dynpro component ZZ_00_BAPINAV) to ZZ_00_CONFIGURE.
2. Adjust the name of the window according to the component name.

Creation of the Configuration Controller and Context Attributes

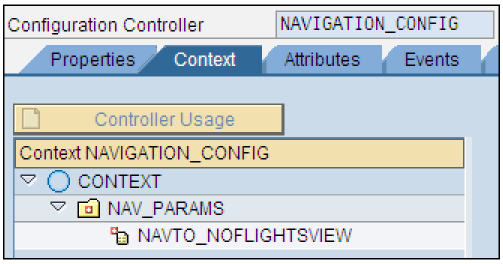
1. The parameter which later should steer the behavior of the application is an explicit parameter and requires a Configuration Controller.
   Create a new custom controller by right clicking the component and select *Create→Custom Controller*. Name it NAVIGATION_CONFIG and save it.



Change the newly created custom controller into a Configuration Controller by right clicking the controller and selecting *Set as Config. Controller*. The icon will change to NAVIGATION_CONFIG .

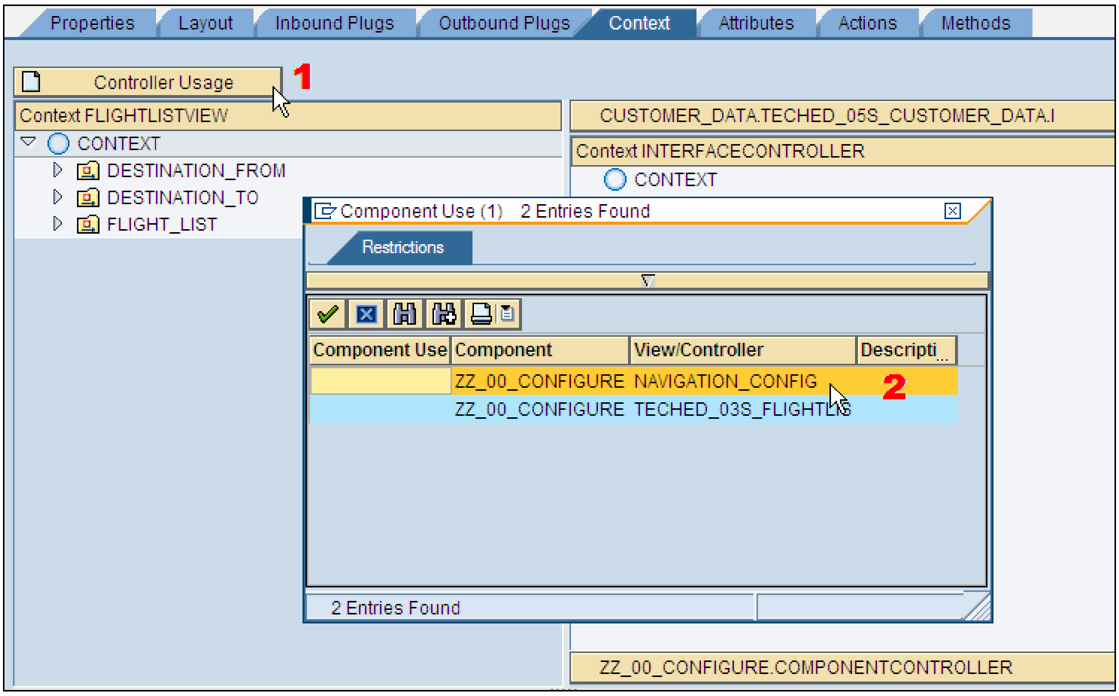


2. Switch to the context tab of the controller NAVIGATION_CONFIG. Create a node NAV_PARAMS (keep all options default) with an attribute NAVTO_NOFLIGHTSVIEW of type WDY_BOOLEAN. The result should look like this:



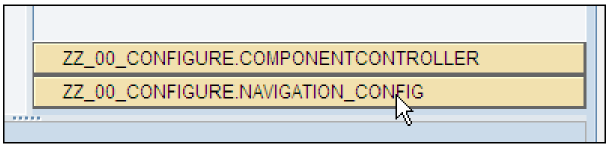
Mapping the Configuration Attribute to the View context

1. To map the configuration context data to the view context switch to the tab *Context* of view FLIGHTLISTVIEW and click on *Controller Usage* (1). Select the controller NAVIGATION_CONFIG from the pop up (2).

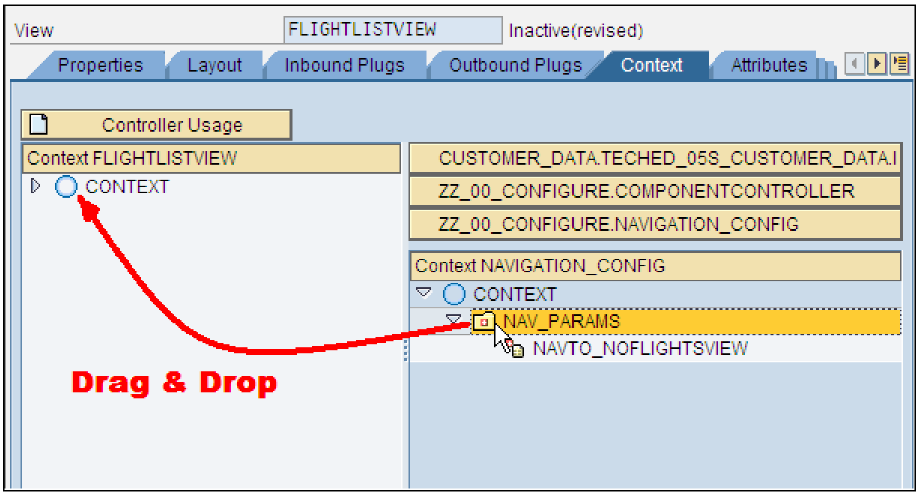


Component Use	Component	View/Controller	Descripti...
	ZZ_00_CONFIGURE	NAVIGATION_CONFIG	
	ZZ_00_CONFIGURE	TECHED_03S_FLIGHTLIS	

To expand the context click on *ZZ_00_CONFIGURE.NAVIGATION_CONFIG*, which just appeared at the right hand side.


ZZ_00_CONFIGURE.COMPONENTCONTROLLER
ZZ_00_CONFIGURE.NAVIGATION_CONFIG

Drag the node NAV_PARAMS of the configuration controller to the context of the view controller and confirm the appearing pop up. The node NAV_PARAMS will appear as a further node in the view context.

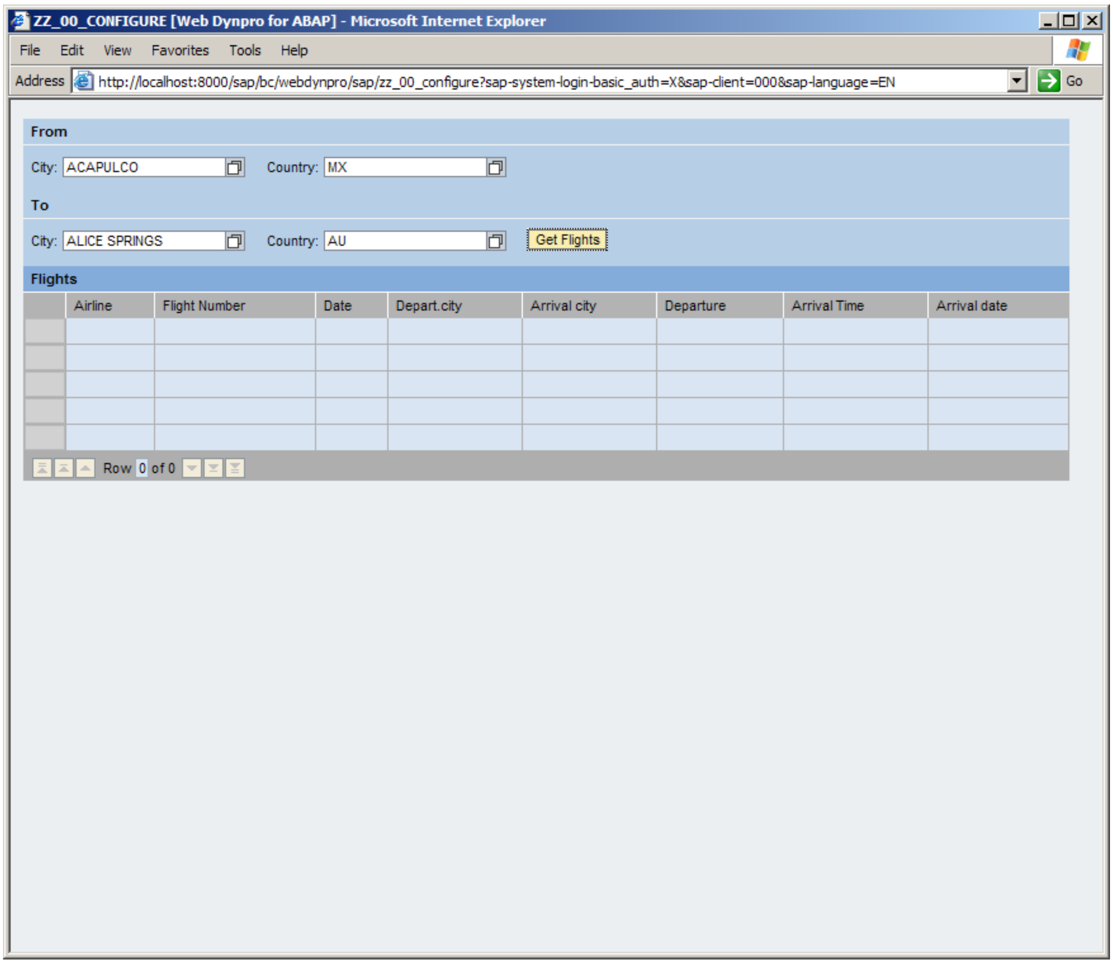


2. Switch to the tab *Methods* of the view FLIGHTLISTVIEW and double click the method *ONACTIONGET_FLIGHTS* to edit the source code.
   Go to the line where the navigation to NOFLIGHTSVIEW is triggered and change the coding, so that the navigation is only triggered if the attribute NAVTO_NOFLIGHTSVIEW of the configuration context is true (= ABAP_TRUE):

```
* get element via lead selection
  Elem_Flight_List = Node_Flight_List->get_Element(  ).

  data: node_nav_params type ref to if_wd_context_node,
        b_navto_noflights type wdy_boolean.

  node_nav_params = wd_context->get_child_node( 'NAV_PARAMS' ).
  node_nav_params->get_attribute( exporting name = 'navto_noflightsview'
                                  importing value = b_navto_noflights ).

* @TODO handle not set lead selection
  if ( Elem_Flight_List is initial and b_navto_noflights = abap_true ).

      wd_This->Fire_No_Flights_Found_Plg(
      ).

  endif.
```

First Test

1. Save and activate everything and test the corresponding application. If you now select ALICE SPRINGS and ACAPULCO, the application should not switch to the view NOFLIGHTSVIEW but should simply show the empty table, because the NAVTO_NOFLIGHTSVIEW attribute is false by default.



ZZ_00_CONFIGURE [Web Dynpro for ABAP] - Microsoft Internet Explorer

http://localhost:8000/sap/bc/webdynpro/sap/zz_00_configure?sap-system-login-basic_auth=X&sap-client=000&sap-language=EN

From

City: ACAPULCO Country: MX

To

City: ALICE SPRINGS Country: AU

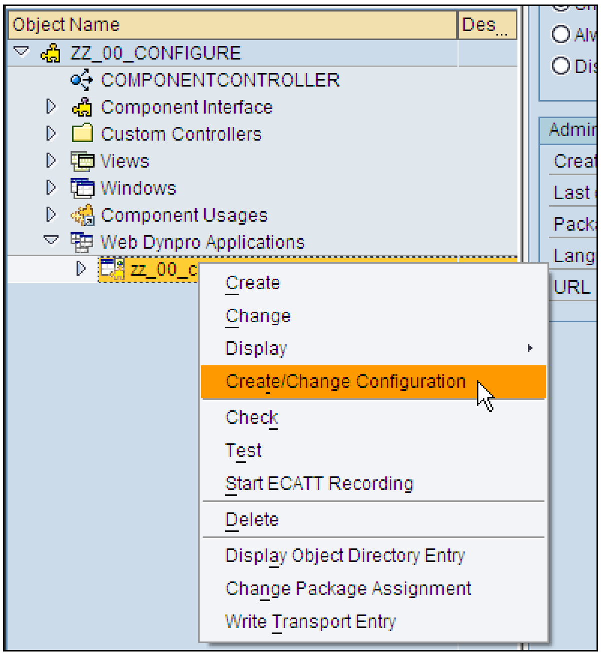
Flights

	Airline	Flight Number	Date	Depart.city	Arrival city	Departure	Arrival Time	Arrival date

Row 0 of 0

Creation of an Application Configuration and corresponding Component Configuration

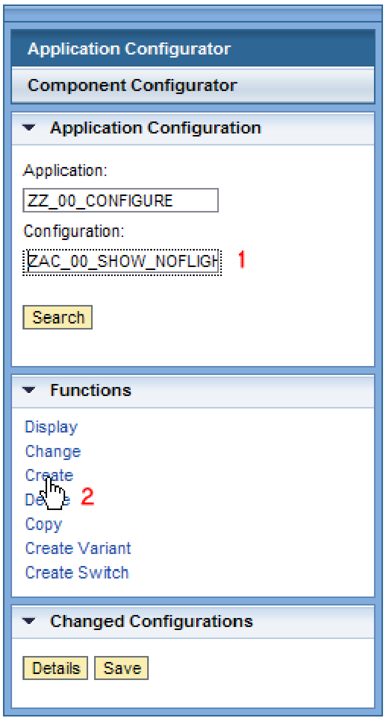
1. Now you should create an application configuration, which sets the attribute NAVTO_NOFLIGHTSVIEW to true, to let the component show the view NOFLIGHTSVIEW.
   Go to the object tree at the left hand side and right click the application object and select *Create/Change Configuration.*



   The configuration editor will pop up in a separate browser window.

   > ⚠ You might get a timeout in the browser window when you start the Application Configurator and Component Configurator for the first times. This is due to background compilation within the NSP system. Just start the Configurators all over again. Another option will be to reset the http timeout parameter (Max. Processing Time) to a higher value.

2. First you have to create an application configuration:
   Enter ZAC_00_SHOW_NOFLIGHT as name for the application configuration (1) and click on *Create* (2)



Application Configurator

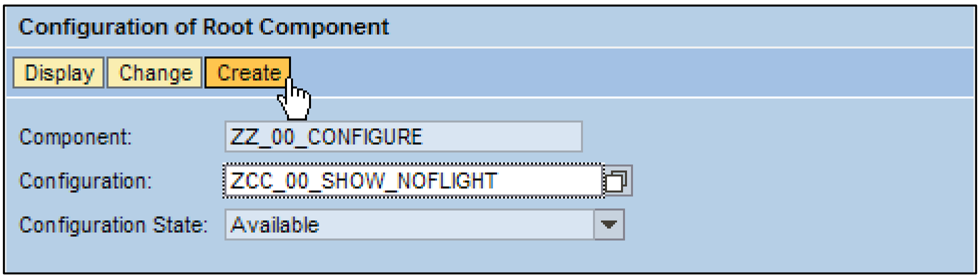
Component Configurator

Application Configuration

Application: ZZ_00_CONFIGURE

Configuration: ZAC_00_SHOW_NOFLIGH

3. Second a component configuration for Web Dynpro component ZZ_00_CONFIGURE is needed. In the appearing box at the right hand side type in *ZCC_00_SHOW_NOFLIGHT* as name for the corresponding component configuration and click *Create*.



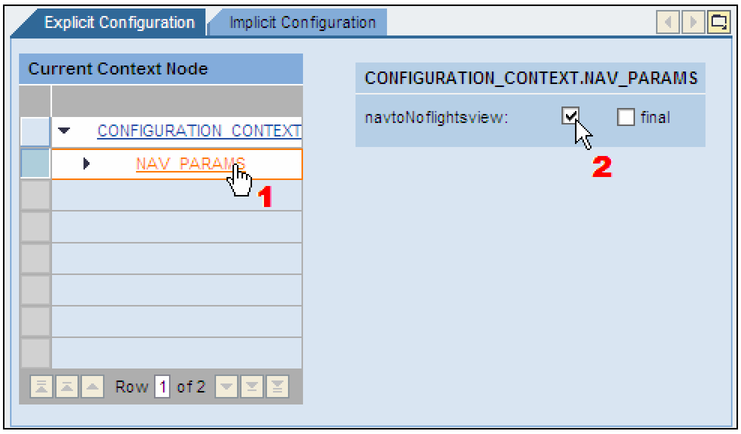
Configuration of Root Component

Component: ZZ_00_CONFIGURE

Configuration: ZCC_00_SHOW_NOFLIGHT

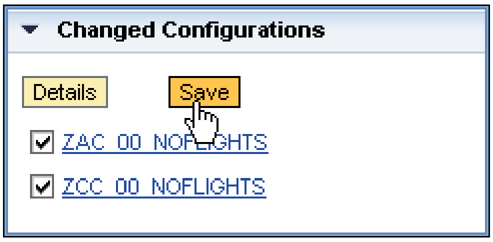
Configuration State: Available

4. Select the attribute NAV_PARAMS (1) and check the left check box (2) which sets the default value for the attribute NAVTO_NOFLIGHTSVIEW in this configuration:



navtoNoflightsview: ☑ □ final

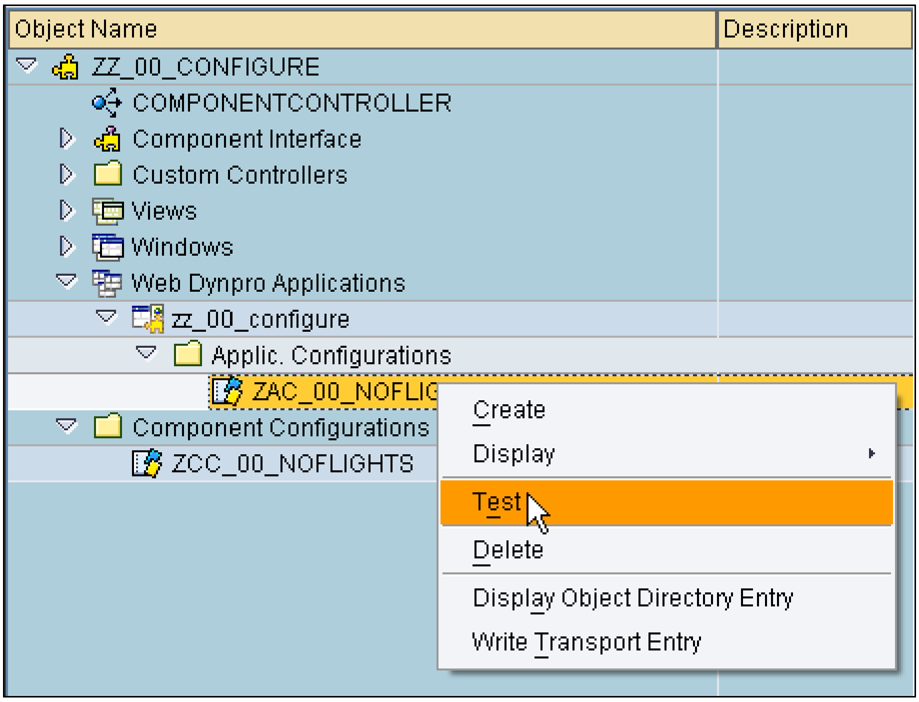
5. Click on *Save* in order to save your configuration settings.



☑ ZAC_00_NOFLIGHTS

☑ ZCC_00_NOFLIGHTS

6. Go back to the ABAP Workbench and click on *Refresh* in the object browser at the left hand side.
   The newly created application and component configurations should appear in the tree. Right click the application configuration and choose *Test*, to start the application with the configuration for displaying the special view if the list of flights is empty:



Test what happens if you select the same cities as before!

7. You may now change the component configuration once more by deselecting the checkbox for attribute NAVTO_NOFLIGHTSVIEW and test the application behaviour once more.

Result

You should now be able to

- create and use a Configuration Controller,
- steer application behaviour depending upon the configuration attribute settings
- use the Application and Component Configurators to set the values for configuration attributes.

SAP Online Help

More information on Configuration, Personalization and Customization can be found at the SAP Help Portal under short link http://help.sap.com/saphelp_nw04s/helpdata/en/22/719f42f2ff7e5fe10000000a155106/frameset.htm or via path help.sap.com → Documentation → SAP NetWeaver → SAP NetWeaver 2004s → English → SAP NetWeaver Library → SAP NetWeaver by Key Capability → Application Platform by Key Capability → ABAP Technology → UI Technology → Web UI Technology → Web Dynpro for ABAP → Web Dynpro for ABAP: Development in Detail → Advanced Concepts → Personalization and Configuration

General information on Web Dynpro for ABAP can be found at the SAP Help Portal under the short link http://help.sap.com/saphelp_nw04s/helpdata/en/77/3545415ea6f523e10000000a155106/frameset.htm or via path help.sap.com → Documentation → SAP NetWeaver → SAP NetWeaver 2004s → English → SAP NetWeaver Library → SAP NetWeaver by Key Capability → Application Platform by Key Capability → ABAP Technology → UI Technology → Web UI Technology → Web Dynpro for ABAP.