Using ALV with a Dynamic Context Node in Web Dynpro for ABAP

Release SAP NetWeaver 2004s

Copyright

© Copyright 2005 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, and Informix are trademarks or registered trademarks of IBM Corporation 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.

Icons in Body Text

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	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`.

Table of Contents

Using an ALV Table with Dynamic Context Nodes

In this tutorial the structure of the context is not known at design time, but at runtime. If an ALV table should be used to display a context node, which is built up at runtime, it is not possible to do external context mapping. This tutorial shows you how to handle the ALV table in combination with dynamic context nodes.

Task

The task of this simple tutorial is to let the user enter the name of a data dictionary table and then create dynamically the context node for this table and display it using the ALV.

Objectives

By the end of this tutorial, you will be able to:

- ✓ Using an ALV with dynamic context nodes

Knowledge

- ☐ Knowledge of ABAP OO programming language
- ☐ Basic knowledge of programming Web Dynpro applications
- ☐ Basic knowledge of ABAP workbench

Create a new Web Dynpro Component

This tutorial starts with the creation of a new Web Dynpro component called **Z00_WDT_FLIGHTLIST_DYN**.

Procedure

Start the ABAP Workbench (*se80*) and create the new Web Dynpro component *Z00_WDT_FLIGHTLIST_DYN*. Use *MAIN* as window name.

View for Entering DDIC Table

Inside the Web Dynpro component the view *MAIN_VIEW* is going to be created and the context is being set up for holding the input of the tablename.

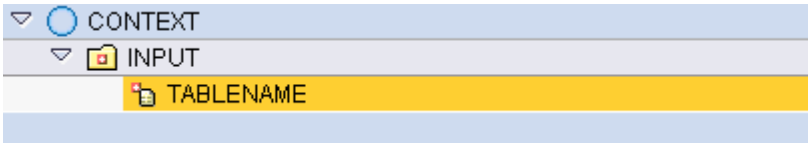
Procedure

Create view *MAIN_VIEW.*

Create view *MAIN_VIEW*.

Create context element and layout for storing DDIC table entry.

In the view *MAIN_VIEW* create a context node *INPUT* with an attribute *TABLENAME* of type string.



Navigate to the layout tab and create the following elements:

- Group *INPUTGROUP* with caption text “Your Input”
- A label and an inputfield for the context attribute *TABLENAME*. Use text “Name of a DDIC table” for the label
- Button *SHOW* with text “Show table” and action *SHOW*
- View container UI element *CONTAINER*

Creating the Dynamic Context Node

Because we do not know the name and structure of the table before the user enters it at runtime, it is not possible to create the context node for the table at design time. Therefore we are now creating the coding for the dynamic table and context node creation.

Navigate into the event handler method *ONACTIONSHOW* of view *MAIN_VIEW* and create the dynamic context node *DYNAMIC_NODE* directly under the context root node.

ONACTIONSHOW()

```
METHOD onactionshow .

  DATA:
*   Node Info
    rootnode_info TYPE REF TO if_wd_context_node_info,

*   Context Nodes
    dyn_node      TYPE REF TO if_wd_context_node,
    tabname_node  TYPE REF TO if_wd_context_node,

*   String (for table name)
    tablename     TYPE string.

* get node info of context root node
  rootnode_info = wd_context->get_node_info( ).

* Get the name of the table to be created
  tabname_node = wd_context->get_child_node( name = 'INPUT' ).
  tabname_node->get_attribute( EXPORTING name = 'TABLENAME'
                               IMPORTING value = tablename ).
  TRANSLATE tablename TO UPPER CASE.

* create sub node named TEST1 of structure (tablename)
  cl_wd_dynamic_tool=>create_nodeinfo_from_struct(
    parent_info = rootnode_info
    node_name = tablename
    structure_name = tablename
    is_multiple = abap_true ).

[…]
```

The next step after having dynamically created the context node is to read the content of the database table and to bind it to the context node.

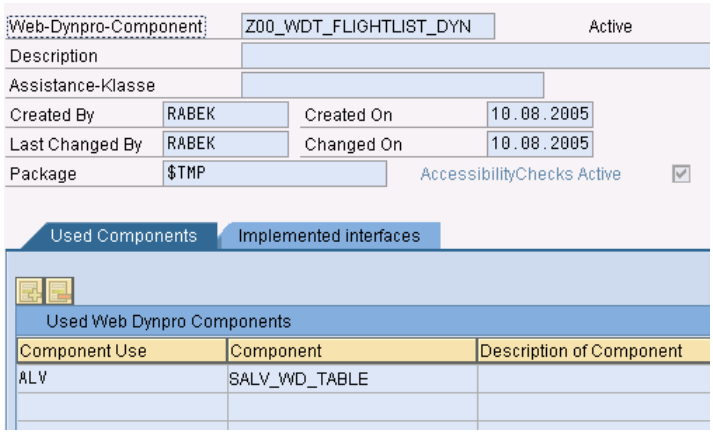
ONACTIONSHOW()

```
[…]

  DATA: stru_tab TYPE REF TO data.

  FIELD-SYMBOLS:
    <tab> TYPE table.

* create internal table
  CREATE DATA stru_tab TYPE TABLE OF (tablename).
  ASSIGN stru_tab->* TO <tab>.

* Get table content
  SELECT * FROM (tablename) INTO CORRESPONDING FIELDS OF TABLE <tab>.

* get instance of new node
  dyn_node = wd_context->get_child_node( name = tablename ).

* Bind internal table to context node.
  dyn_node->bind_table( <tab> ).
```

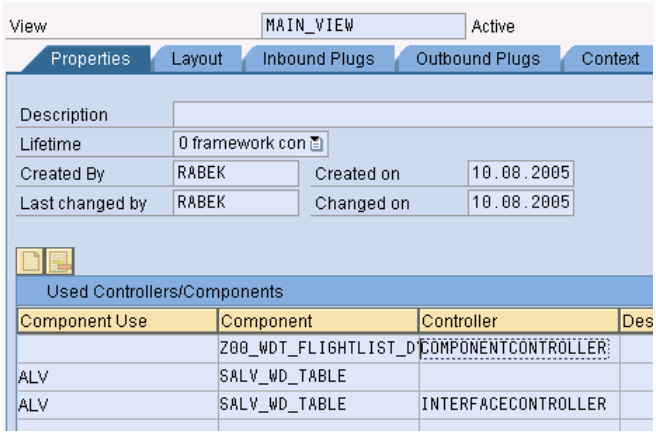
```
[…]
```

Define component usage SALV_WD_TABLE in view *MAIN_VIEW.*

Declare the usage of the ALV component inside component *Z00_WDT_FLIGHTLIST_DYN:*



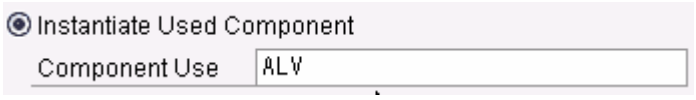
Declare the usage of the ALV component inside view MAIN_VIEW:



Set Data to ALV for display (via ALV component interface method).

Instantiate the ALV component.

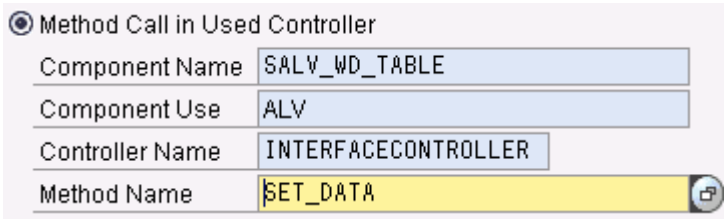
Instantiate the ALV component. Use the Web Dynpro code wizard:



Call interface method SET_DATA( ).

Pass the context node to the ALV (alternatively to the external context mapping for static context nodes).

Use the Web Dynpro code wizard:



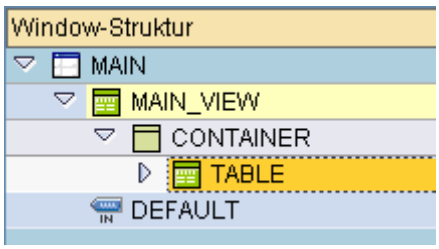
ONACTIONSHOW()

```
[…]

* Instantiate ALV component
  DATA: l_ref_cmp_usage TYPE REF TO if_wd_component_usage.

  l_ref_cmp_usage =   wd_this->wd_cpuse_alv( ).
  IF l_ref_cmp_usage->has_active_component( ) IS INITIAL.
    l_ref_cmp_usage->create_component( ).
  ENDIF.


* Pass context node to ALV
  DATA: l_ref_interfacecontroller TYPE REF TO iwci_salv_wd_table .

  l_ref_interfacecontroller = wd_this->wd_cpifc_alv( ).
  l_ref_interfacecontroller->set_data( dyn_node ).


 […]
```

Embed view TABLE of component SALV_WD_TABLE into own window.

It's not possible to display a view inside a browser, but a window. Therefore the just created *MAIN_VIEW* and the view *TABLE* of the ALV component have to be assigned to the window *MAIN*.

Navigate to the window and into tab window and embed view *MAIN_VIEW* into the window. Thereafter embed view *TABLE* of the ALV component into the view container ui element *CONTAINER*.



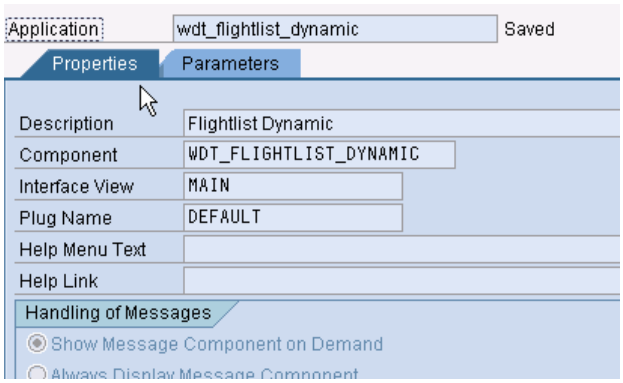
Activate your Web Dynpro component.

Create and Test Web Dynpro Application

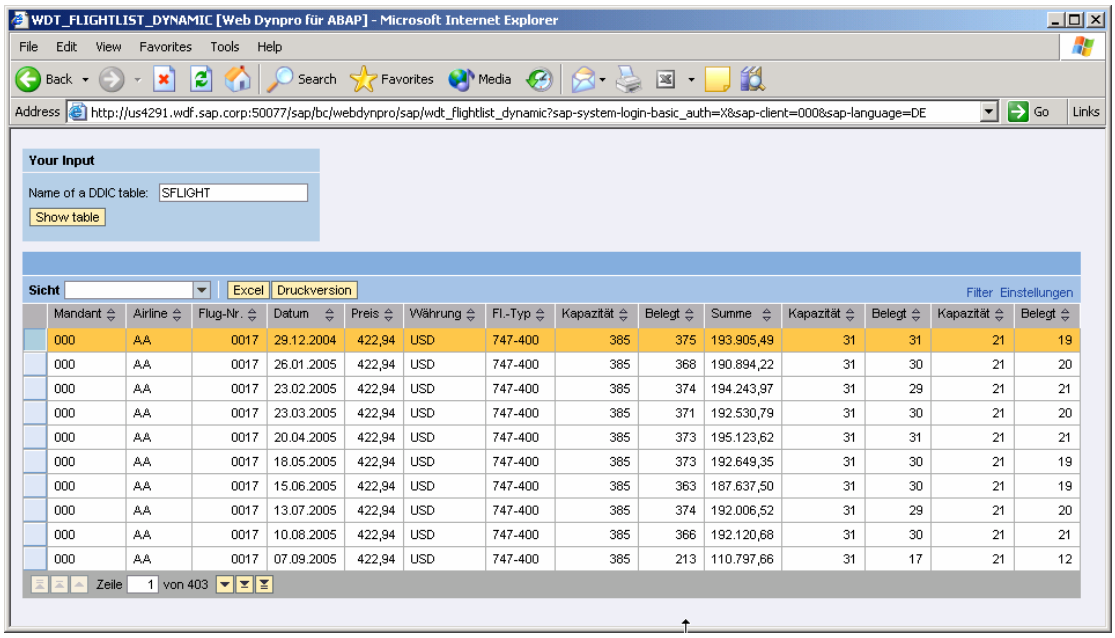
Each Web Dynpro component needs a Web Dynpro application to be executed.

Procedure

Create a Web Dynpro application for your Web Dynpro component:



Test your Web Dynpro application. The result looks like the following:



Author Bio

Claudia Dangers is a senior development consultant in SAP's Software Technology and Development department. Since she joined SAP in 1999 she has worked on numerous projects and gained practical experience in ABAP and BSP development, in the creation of concepts, in coaching and code reviews, and as a sub-project lead and training instructor. Claudia is very interested in new technologies. Currently she is dealing with Web Dynpro ABAP, kernel-based BADI's and the Switch and Enhancement Framework.