

Editing ALV 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
<i>Example text</i>	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

Copyright.....	2
Icons in Body Text	3
Typographic Conventions.....	3
Table of Contents	4
Task	5
Objectives.....	5
 Copying an Existing Web Dynpro Component	6
Procedure.....	6
 Modify View ResultView for Displaying ALV Table.....	6
Configure ALV in ResultView to make ALV editable.	6
 Save Changes	7
Procedure.....	7
 Test Your Web Dynpro Application	9
<i>Author Bio</i>.....	10



Handling of Editable ALV Tables in Web Dynpro

This tutorial shows you how to make a column of your ALV table editable.

Task

The starting point of this tutorial is the solution application of the tutorial, “Programming the ALV Configuration Model in Web Dynpro for ABAP.”

The task of this tutorial is to configure the ALV to make column “price” editable.

Objectives

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

- ✓ Make a column of the ALV table editable

Knowledge

- Knowledge of ABAP OO programming language
- Basic knowledge of programming Web Dynpro applications
- Basic knowledge of ABAP workbench
- Knowledge of the tutorial “Simple example for using the ALV inside Web Dynpro for ABAP”.
- Knowledge of the tutorial “Programming the ALV configuration model inside Web Dynpro for ABAP”



Copying an Existing Web Dynpro Component

In the system there is a master copy of a Web Dynpro component called **WDT_FLIGHTLIST_CONFIG**. You can copy this component as described below.

Procedure

Copying the Web Dynpro Component

1. Start the ABAP Workbench (se80) and select the Web Dynpro component **WDT_FLIGHTLIST_CONFIG**.
2. Open the context menu of **WDT_FLIGHTLIST_CONFIG** and copy the Web Dynpro component to **Z00_WDT_FLIGHTLIST_EDIT**.
3. Open the context menu of the new component **Z00_WDT_FLIGHTLIST_EDIT** and create a Web Dynpro application **Z00_WDT_FLIGHTLIST_EDIT**.
4. Select the interface view by using F4 help. Choose **MAIN**.
5. Select a plug name by using F4 help and choose **default**.
6. Activate the new Web Dynpro component.



Modify View ResultView for Displaying ALV Table

In this tutorial we are changing the ALV table from tutorial "Programming the ALV Configuration Model in Web Dynpro for ABAP" by making column "price" editable.

Configure ALV in ResultView to make ALV editable.

Set table editable.

First it is necessary to use an input field as cell editor for the column "price," which should be editable. Therefore we need to enhance method **WDDOINIT** of view **ResultView**:

```

WDDOINIT ( )
[...]
```

```

* set cell editor for input fields (~make column PRICE editable)
DATA: lr_column_settings TYPE REF TO if_salv_wd_column_settings,
      lr_input_field     TYPE REF TO cl_salv_wd_uie_input_field.

lr_column_settings ?= l_value.
lr_column = lr_column_settings->get_column( 'PRICE' ).

CREATE OBJECT lr_input_field EXPORTING value_fieldname = 'PRICE'.
lr_column->set_cell_editor( lr_input_field ).
[...]
```

Set table editable.

Additionally the “read only mode” has to be set to `abap_false` to make the table editable.

```

WDDOINIT( )
[...]
* set read only mode to false (and display edit toolbar)
data: lr_table_settings type ref to if_salv_wd_table_settings.

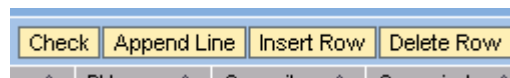
lr_table_settings ?= l_value.
lr_table_settings->set_read_only( abap_false ).

ENDMETHOD.

```

Without this setting the input field will not be editable!

In addition this setting displays the edit toolbar:



Save Changes

A save button has to be implemented to save the user changes to the database.

Hint: In this tutorial the save functionality is only simulated to not change the flight data model content.

Procedure

Create button SAVE.

Place a save button on view RESULTVIEW with name `BTN_SAVE`.
Enter “Save” into the text property.

The screenshot shows the SAP Web Dynpro development environment. The view 'RESULTVIEW' is active. The 'Layout' tab is selected, showing a 'ViewContainer:CONTAINER' with a 'Save' button. The 'Properties (Button)' table is visible, showing the configuration for the 'BTN_SAVE' button.

Property	Value
Properties (Button)	
ID	BTN_SAVE
Layout Data	MatrixHeadData
design	standard
enabled	
explanation	
imageFirst	
imageSource	
text	Save

Implement action handler SAVE.

Create action SAVE in the “onAction” property of button BTN_SAVE.



In action handler method ONACTIONS SAVE the interface method DATA_CHECK of the ALV component is invoked. If changes have been made then the interface event ON_DATA_CHECK is raised.

```

ONACTIONS SAVE ( )
METHOD onactionsave.

DATA: l_ref_interfacecontroller TYPE REF TO iwci_salv_wd_table .

* Check for changes
l_ref_interfacecontroller = wd_this->wd_cpifc_alv( ).
l_ref_interfacecontroller->data_check( ).

ENDMETHOD.

```

Implement event handler ONDATA CHECK.

Implement an event handler method for the interface event ON_DATA_CHECK with the following functionality:

- Get actual/changed content of the flight table
- Update DB
- Display a success message

```

ONDATA CHECK ( )
METHOD ondatacheck .

DATA: node_node_flighttab TYPE REF TO if_wd_context_node,
      elem_node_flighttab TYPE REF TO if_wd_context_element,
      lt_sflight TYPE if_resultview=>elements_node_flighttab.

* save data only if no error has occurred
CHECK r_param->t_error_cells IS INITIAL.

* navigate from <CONTEXT> to <NODE_FLIGHTTAB> via lead selection
node_node_flighttab
  = wd_context->get_child_node( name = `NODE_FLIGHTTAB` ).

* get data from context node <NODE_FLIGHTTAB>
node_node_flighttab->get_static_attributes_table(
  IMPORTING table = lt_sflight ).

* save data to database
* update... => only simulate, to not change the flight data model
* content!

* Create success message
* get message manager
DATA: l_current_controller TYPE REF TO if_wd_controller,
      l_message_manager TYPE REF TO if_wd_message_manager.

l_current_controller ?= wd_this->wd_get_api( ).

```

```

CALL METHOD l_current_controller->get_message_manager
  RECEIVING
    message_manager = l_message_manager.

* report message
CALL METHOD l_message_manager->report_success
  EXPORTING
    message_text = 'Data was successfully saved.'.

ENDMETHOD.

```



Test Your Web Dynpro Application

The result will look like the following:

The screenshot shows a web browser window titled 'Flightlist Edit - Microsoft Internet Explorer'. The address bar shows the URL: `http://us4293.wdf.sap.corp:50077/sap/bc/webdynpro/sap/wdt_flightlist_edit?sap-system-login-`. The page content includes a search form with 'Airline: LH' and 'Flight Number: 0400'. Below the search form is a table with columns: Airline, Flight No., Date, Price, Currency, Pl.type, Capacity, Occupied, and Total. The table contains five rows of flight data. A red box highlights the 'Microsoft Excel' button above the table. Another red box highlights the table itself. A third red box highlights the 'Save' button at the bottom left of the page. The 'Save' button is being clicked by a mouse cursor.

Airline	Flight No.	Date	Price	Currency	Pl.type	Capacity	Occupied	Total
LH	0400	09.07.2005	99999,99	EUR	A310-300	280	○△○	208.977,48
LH	0400	19.03.2005	666,00	EUR	A310-300	280	○△○	211.441,68
LH	0400	14.05.2005	666,00	EUR	A310-300	280	○△○	209.450,34
LH	0400	06.08.2005	666,00	EUR	A310-300	280	○△○	210.735,72
LH	0400	28.09.2005	666,00	EUR	A310-300	280	○△○	212.853,60

Data was successfully saved. 1 Message [Show List](#)

Select the Flights

Airline:

Flight Number:

View [Filter Settings](#)

Airline	Flight No.	Date	Price	Currency	Pl.type	Capacity	Occupied	Total
LH	0400	09.07.2005	99.999,99	EUR	A310-300	280	0△0	208.977,48
LH	0400	19.03.2005	666,00	EUR	A310-300	280	0△0	211.441,68
LH	0400	14.05.2005	666,00	EUR	A310-300	280	0△0	209.450,34
LH	0400	06.08.2005	666,00	EUR	A310-300	280	0△0	210.735,72
LH	0400	28.09.2005	666,00	EUR	A310-300	280	0△0	212.853,60

Row 1 of 17

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.