Using Events with ALV Tables
in Web Dynpro for ABAP

Release SAP NetWeaver 2004s
Icons in Body Text

<table>
<thead>
<tr>
<th>Icon</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>⚠️</td>
<td>Caution</td>
</tr>
<tr>
<td>🔍</td>
<td>Example</td>
</tr>
<tr>
<td>ⓘ</td>
<td>Note</td>
</tr>
<tr>
<td>💡</td>
<td>Recommendation</td>
</tr>
<tr>
<td>_syntax</td>
<td>Syntax</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Type Style</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Example text</em></td>
<td>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.</td>
</tr>
<tr>
<td><em>Example text</em></td>
<td>Emphasized words or phrases in body text, graphic titles, and table titles.</td>
</tr>
<tr>
<td><strong>EXAMPLE TEXT</strong></td>
<td>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.</td>
</tr>
<tr>
<td><em>Example text</em></td>
<td>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.</td>
</tr>
<tr>
<td><em>Example text</em></td>
<td>Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.</td>
</tr>
<tr>
<td><code>&lt;Example text&gt;</code></td>
<td>Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.</td>
</tr>
<tr>
<td><strong>EXAMPLE TEXT</strong></td>
<td>Keys on the keyboard, for example, F2 or ENTER.</td>
</tr>
</tbody>
</table>
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

Set CellEditor of a column in the ALV table to an editor that triggers an
ON_CLICK event ......................................................................................................... 6
CellEditors which are not editable (editable cell editors trigger ON_DATA_CHANGED event)
and allow actions to be performed on them (Button, ToggleButton, LinkToAction) trigger the
ON_CLICK event. Therefore it is necessary to set a column cell editor to one of these cell editors
in order to trigger this event. ..................................................................................... 6
Use a button for column CARRID. ............................................................................. 6
Use a link for column CONNID.................................................................................. 7

Display Event Information .................................................................................. 7
Create new context node............................................................................................ 7
Create event handler method onClick........................................................................... 8
Display the event......................................................................................................... 9

Test Your Web Dynpro Application ....................................................................... 9

Author Bio ............................................................................................................. 10
Using Events with ALV Tables in Web Dynpro

This tutorial shows you how to use events inside an ALV table in Web Dynpro.

Task

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

The task of this tutorial is to use the ON_CLICK event. This event is triggered when an action on a column holding a non editable cell editor (Button, ToggleButton, LinkToAction) is performed (ButtonClick, LinkClick).

Objectives

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

✅ Display additional information, if a cell is selected

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_EVENT.
3. Open the context menu of the new component Z00_WDT_FLIGHTLIST_EVENT and create a Web Dynpro application Z00_WDT_FLIGHTLIST_EVENT.
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.

Set CellEditor of a column in the ALV table to an editor that triggers an ON_CLICK event

CellEditors which are not editable (editable cell editors trigger ON_DATA_CHANGED event) and allow actions to be performed on them (Button, ToggleButton, LinkToAction) trigger the ON_CLICK event. Therefore it is necessary to set a column cell editor to one of these cell editors in order to trigger this event.

Use a button for column CARRID.

One example for a clickable UI element is a button. We are going to use a button cell editor for the column CARRID. When the button is clicked on the event ON_CLICK will be triggered.

Navigate on the methods tab of view ResultView and add the following code at the end of method WDDOINIT:

```
WDDOINIT()

* Display button in column carrid
DATA: lr_button TYPE REF TO cl_salv_wd_uie_button.
  lr_column = l_value->if_salv_wd_column_settings~get_column( 'CARRID' ).
  CREATE OBJECT lr_button.
  lr_button->set_text_fieldname( 'CARRID' ).
  lr_column->set_cell_editor( lr_button ).
endmethod.
```
Use a link for column CONNID

Another example for a clickable UI element is a link. Be careful: There are two kinds of links available: “link to url” and “link to action”. Only the “link to action” is able to trigger an event in the ALV. Therefore we are going to use a “link to action” cell editor for the column CONNID. When a link in this column is clicked on the event ON_CLICK will be triggered.

Navigate on the methods tab of view ResultView and add the following code at the end of method WDDOINIT:

```abap
WDDOINIT()
[
  * Display link in column connid
  DATA: lr_link TYPE REF TO cl_salv_wd_uie_link_to_action.
  lr_column = l_value->if_salv_wd_column_settings~get_column( 'CONNID' ).
  CREATE OBJECT lr_link.
  lr_link->set_text_fieldname( 'CONNID' ).
  lr_column->set_cell_editor( lr_link ).
endmethod.
```

Display Event Information

The event that is raised e.g. through a button click in the ALV table shall be displayed in a second table on the RESULTVIEW.

Create new context node.

First we need a new context node to hold the information on the last event.

Navigate on the context tab of view ResultView and add the new node EVENT_PROPERTIES. Set the cardinality to 0..n.

Add the attributes NAME and VALUE (both of type string) to the newly created context node EVENT_PROPERTIES.
Create event handler method `onClick`.

Second we need to store the event information in the new context node `EVENT_PROPERTIES`.

Navigate to the methods tab of view `ResultView` and create event handler method `onClick` for the `On_CLICK` event of the ALV component:

Implement event handler method `ONCLICK` with the following functionality:

- Fill an internal table with the following information: selected column, selected index, attribute name and attribute value
- Bind the internal table to context node `EVENT_PROPERTIES`

```plaintext
ONCLICK()
method ONCLICK.

  data: lr_node type ref to if_wd_context_node,
  lt_event_properties type if_wd_context=>event_properties,
  ls_event_properties type if_wd_context=>element_event_properties.
  field-symbols: <l_value> type any.
  * fill internal table
    ls_event_properties-name  = 'COLUMN_ID',
    ls_event_properties-value = r_param->column.
    append ls_event_properties to lt_event_properties.

    ls_event_properties-name  = 'INDEX',
    ls_event_properties-value = r_param->index.
    append ls_event_properties to lt_event_properties.

    ls_event_properties-name  = 'ATTRIBUTE',
    ls_event_properties-value = r_param->attribute.
    append ls_event_properties to lt_event_properties.

    assign r_param->value->value to <l_value>.
    ls_event_properties-name  = 'VALUE',
    ls_event_properties-value = <l_value>.
    append ls_event_properties to lt_event_properties.

  * navigate to context node `EVENT_PROPERTIES`
    lr_node = wd_context->get_child_node( 'EVENT_PROPERTIES' ).

  * bind internal table to context node
```
lr_node->bind_table( lt_event_properties ).
endmethod.

Display the event.

The additional information, regarding the event properties, should be displayed in a WD table control. In the tabstrip “Layout” of the view RESULTVIEW, add a table control to the view elements. Use the context menu to bind the data node EVENT_PROPERTIES to the table control. This will automatically create the required columns and their bindings.

Test Your Web Dynpro Application

The result will look like the following:
SAP List Viewer (ALV) in Web Dynpro Applications

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.