How to Implement Multi Value Paste for Select Options 1.0
APPLIES TO:
SAP NetWeaver 7.0, 7.01 and 7.02

SUMMARY
This document describes step-by-step how you can enable multi value paste for the select options 1.0 for SAP NetWeaver 7.0, 7.01 and 7.02 using the enhancement framework. In addition, it describes how multi value paste works for SAP NetWeaver 7.31 SP07 and 7.40 SP2, where it is automatically available.

Authors: Michael Kraeft, SAP TIP Core UI Development
Lisa Monshausen, SAP TIP KM
Company: SAP AG
Created on: 27.02.2013
TABLE OF CONTENTS

APPLIES TO: .................................................................................................................. 2
SUMMARY ..................................................................................................................... 2
INTRODUCTION ............................................................................................................. 4

MULTI VALUE PASTE IN SELECT OPTIONS 1.0 .......................................................... 4
Multi Value Paste on the Selection Screen .................................................................... 4
Multi Value Paste on the Extended Options Screen ..................................................... 5

HOW TO IMPLEMENT MULTI VALUE PASTE FOR SAP NETWEAVER 7.0, 7.01 OR 7.02 .... 5
1. Creating an Enhancement Implementation for Component WDR_SELECT_OPTIONS .... 6
2. Defining a New Attribute for the Component Controller ............................................ 7
3. Creating a New View for the Value Paste ................................................................. 8
   3.1 Creating an Additional View in Enhancement Mode .................................................. 8
   3.2 Creating a New Context Attribute ......................................................................... 8
   3.3 Inserting a TextEdit UI element ........................................................................... 9
   3.4 Creating the Data Binding for the TextEdit .............................................................. 9
   3.5 Defining an Additional Attribute for the View Controller ..................................... 9
   3.6 Implementing an Event Handler Method for the OK Button ................................... 9
   3.7 Assigning an Action to the OK Button .................................................................. 12
4. Creating a Window to Embed the New View into .................................................... 13
5. Enhancing the EXTENDED_OPTIONS View ............................................................ 13
   5.1 Inserting a ToolBarLinkToAction UI Element ....................................................... 13
   5.2 Creating an onAction Event Handler for the ToolBarLinkToAction ................... 14
   5.3 Creating a Post-Exit in WDDOMODIFYVIEW of View EXTENDED_OPTIONS .......... 15
6. (Optional) Influencing the Availability of the Multi-Value Paste Functionality ........... 16
7. Testing your Enhancement Implementation ............................................................. 17

RELATED CONTENT ....................................................................................................... 18
INTRODUCTION

Multi value paste allows users of your application to copy several values from a table or a text editor and insert them into the select options so that they can be used for searching. Users can use the standard copy and paste clipboard functions (CTRL-C and CTRL-V).

For select options 1.0, multi value paste is available as of SAP NetWeaver 7.31 SP07 and 7.40 SP2. If you are using a lower release, however, you need to implement an enhancement for the old select options component to enable this functionality. This document describes how multi value paste works for SAP NetWeaver 7.31 SP07 and 7.40 SP2. In addition, it explains step-by-step how you enable this functionality for lower releases (SAP NetWeaver 7.0, 7.01 and 7.02).

MULTI VALUE PASTE IN SELECT OPTIONS 1.0

Starting with SAP NetWeaver 7.31 SP07 and 7.40 SP2, we offer multi value paste in the select options 1.0. This functionality is automatically available and cannot be switched off.

There are two possibilities to insert several entries via the clipboard: on the selection screen or on the extended options screen. If the operator is_between is available, two columns can be pasted.

Multi Value Paste on the Selection Screen

The graphic below displays the process of multi value paste on the selection screen:
Only the first two columns of the clipboard are evaluated and pasted. The rest of the columns is ignored. The first column is pasted into the low field and the second into the high field. The values assigned before are removed.

The content is checked according to the type. If errors occur, the extended options popup is called and the pasted data is shown (including wrong entries). The user can correct the errors. If the user chooses Cancel on the popup, the former values are revoked.

Multi Value Paste on the Extended Options Screen

The graphic below displays the process of multi value paste on the extended options screen:

On the extended options screen, the exact position of the cursor is the starting point for the paste. The pasted values replace the values in the line where the paste takes place and the lines below. The values in the lines above the paste position remain. If the paste is executed on a high field, only the upper values are changed.

You can find a test application in the system under WDR_TEST_SEL_OPT_AUTO_10.

HOW TO IMPLEMENT MULTI VALUE PASTE FOR SAP NETWEAVER 7.0, 7.01 OR 7.02

To enable multi value paste for SAP NetWeaver 7.0, 7.01 or 7.02, you use the enhancement framework to enhance the select options component WDR_SELECT_OPTIONS.

This workaround only allows you to offer multi value paste for the Multiple Selection dialog box.
As displayed in the graphic below, a link is available on the *Multiple Selection* dialog box. This link opens another dialog box, in which users can enter the values they want to paste. If the user presses *OK*, the values are transferred to the *Multiple Selection* dialog box.

The following graphic displays the process of multi value paste once you have implemented the enhancement:

The following sections describe the procedure of implementing the enhancement.

1. **Creating an Enhancement Implementation for Component WDR_SELECT_OPTIONS**

First of all, you have to create an enhancement implementation for component `WDR_SELECT_OPTIONS`. You have to choose this implementation in the *Select or Create Enhancement Implementation* dialog box, which appears each time you want to make changes to `WDR_SELECT_OPTIONS` in enhancement mode (the individual changes are described in steps 2-7).

1) In the object list in `SE80`, open component `WDR_SELECT_OPTIONS`. Make sure you are in display mode.

2) In the Web Dynpro Explorer toolbar, choose (Enhance) to continue in enhancement mode.

3) In the dialog box that appears, enter a name (e.g. `ZZ_ENH_SEL_OPT`) and a short text for your enhancement implementation.

4) Choose *Creation of Enhancement* (Enter).

5) Assign a package and choose *Save* (Enter).
The new enhancement implementation is displayed in a separate node in the object list under the component node:

You can now make the required enhancements.

2. **Defining a New Attribute for the Component Controller**

In this step, you define an additional attribute for the component controller. This attribute is used to store a reference to the dynamically created context node containing the values of the extended option table. This allows you to share data between the existing view (EXTENDED_OPTIONS) and the view which you create in step 3.

1. Select the **COMPONENT CONTROLLER** node in the component object list.
2. Open the **Attributes** tab.
3. In the **Attribute** column, enter a name for the new attribute (e.g. MO_CONTEXT_NODE).
4. Enter **IF_WD_CONTEXT_NODE** as **Associated Type**.
3. Creating a New View for the Value Paste

In the following steps, you create the view that should appear as a popup when users select the *Multi Value Paste* link. This view should contain a `TextEdit` UI element to allow users to paste their values:

![Image](image.png)

3.1 Creating an Additional View in Enhancement Mode

1) Select the **Views** node in the component object list.
2) In the context menu, choose **Create as Enhancement**.
3) Enter a name for the new view (e.g. **MULTI_VALUE**).

This view is now listed in the **Enhancement Implementations** node:

![View List](view_list.png)

3.2 Creating a New Context Attribute

The value of the `TextEdit` should be assigned by user input. Therefore, you need to create a new context attribute and bind it to the property `value` of the `TextEdit` (cf. step 3.4).

1) Open the **Context** tab of the view you have just created.
2) Select the root node **CONTEXT**, and choose **Create** → **Attribute**.
3) Enter a name for the new context attribute (e.g. **MV_STR**).
3.3 Inserting a TextEdit UI element

1) Open the **Layout** tab.
2) In the context menu of **ROOTUIELEMENTCONTAINER** choose the entry **Insert element**.
3) Enter an ID for the element (e.g. MV) and choose **TextEdit** from the input help for the **type** line.

![TextEdit UI element](image)

3.4 Creating the Data Binding for the TextEdit

To bind the property **value** of the **TextEdit** to the context attribute you created in step 3.2, proceed as follows:

![Data binding configuration](image)

3.5 Defining an Additional Attribute for the View Controller

In this step, you define an additional attribute for the controller of your new view. This attribute is used as a reference to the popup window containing the new view.

1. Open the **Attributes** tab.
2. In the **Attribute** column, enter a name for the new attribute (e.g. MO_WINDOW).
3. Enter **IF_WD_WINDOW** as **Associated Type**.

3.6 Implementing an Event Handler Method for the OK Button

If users choose **OK** on the dialog box that appears when they navigate to the **Multi Value Paste** link, a method should be called. In this method, you define that the data from the clipboard is transferred. These values are converted from external to internal representation. An error message should be displayed if a wrong value was entered.

1) Open the **Actions** tab.
2) In the **Action** column, enter a name (e.g. ON_MV_OK):
3) Double-click on the new entry to implement the event handler method. You can use the following code as a template and adapt it to your own needs:

```abap
method ONACTION_ON_MV_OK.

DATA lo_el_context TYPE REF TO if_wd_context_element.
DATA ls_context TYPE wd_this->Element_context.
DATA lv_mv_str TYPE wd_this->Element_context-mv_str.
DATA lv_close_me TYPE wdy_boolean VALUE abap_true.

* get element via lead selection
  lo_el_context = wd_context->get_element( ).
* @TODO handle not set lead selection
  IF lo_el_context IS INITIAL.
  ENDIF.

* get single attribute
  lo_el_context->get_attribute( EXPORTING
    name = 'MV_STR'
    IMPORTING
    value = lv_mv_str ).

check lv_mv_str is not initial.

DATA: str_tab TYPE string_table.
DATA: lv_line TYPE string.
DATA: lv_index TYPE i.

DATA: lo_ele TYPE ref to if_wd_context_element.
DATA: lo_attr_info TYPE WDR_CONTEXT_ATTRIBUTE_INFO.
DATA: lo_node_info TYPE REF TO if_wd_context_node_info.
DATA: ld_data TYPE REF TO data.
FIELD-SYMBOLS: <cont> TYPE any.

lo_node_info = wd_comp_controller->mo_context_node->GET_NODE_INFO( ).
lo_attr_info = lo_node_info->GET_ATTRIBUTE( 'LOW' ).
CREATE DATA ld_data TYPE HANDLE lo_attr_info-rtti.
ASSIGN ld_data->* to <cont>.

* num_lines = wd_comp_controller->mo_context_node->get_element_count( ).

DATA: lo_exc TYPE REF TO cx_root.
SPLIT lv_mv_str AT CL_ABAP_CHAR_UTILITIES->NEWLINE INTO TABLE str_tab.
```
How to Implement Multi Value Paste for Select Options 1.0

```
loop at str_tab into lv_line.
  lv_index = sy-tabix.
  lo_ele = wd_comp_controller->mo_context_node->GET_ELEMENT( lv_index ).
  if lo_ele is not bound.
    * create new element via assistance class
      wd_assist->on_ext_opt_insert_line( ).
      lo_ele = wd_comp_controller->mo_context_node->GET_ELEMENT( lv_index ).
  endif.
  try.
    CL_WDR_CONVERSION_UTILS=>FROM_STRING(
      exporting
        IN = lv_line " Input String
        changing
        DATA = <cont> " Date to Be Filled
    )..
    lo_ele->SET_ATTRIBUTE( value = <cont> name = 'LOW' ).
    catch CX_WDR_CONVERSION_EXCEPTION INTO lo_exc. " Conversion Error
      * put message...
    * get message manager
      data lo_api_controller type ref to if_wd_controller.
      data lo_message_manager type ref to if_wd_message_manager.
    lo_api_controller ?= wd_this->Wd_Get_Api( ).
    CALL METHOD lo_api_controller->GET_MESSAGE_MANAGER
      RECEIVING
        MESSAGE_MANAGER = lo_message_manager.
    * report message
      CALL METHOD lo_message_manager->REPORT_EXCEPTION
        EXPORTING
          MESSAGE_OBJECT = lo_exc
          * MSG_USER_DATA =
          * MESSAGE_TYPE = CO_TYPE_ERROR
          * IS_PERMANENT = ABAP_FALSE
          * SCOPE_PERMANENT_MSG = CO_MSG_SCOPE_CONTROLLER
          * VIEW =
          * SHOW AS_POPUP =
          * CONTROLLER_PERMANENT_MSG =
          * MSG_INDEX =
          * CANCEL_NAVIGATION = abap_true
          * ENABLE_MESSAGE_NAVIGATION =
          * COMPONENT =
          * RECEIVING =
          * MESSAGE_ID =
          lv_close_me = abap_false.
      ENDTRY.
  endtry.
endloop.
if lv_close_me = abap_true.
  wd_this->mo_window->close( ).
endif.
endmethod.
```
3.7 Assigning an Action to the OK Button

In this step, you assign the action ON_MV_OK to the OK button. To do this, you implement the SUBSCRIBE_TO_BUTTON_EVENT method of IF_WD_WINDOW in the WDDOMODIFYVIEW hook method.

1. Open the Methods tab.

2. Double-click on WDDOMODIFYVIEW and enter the following code:

```abap
method WDDOMODIFYVIEW.

  data:
      lr_window_controller type ref to if_wd_window_controller,
      lr_window_manager TYPE REF TO if_wd_window.
  * initialize the helper class once
    check first_time = abap_true.

  * request focus (7.00 / 7.01)
    * data: lo_te type ref to IF_WD_VIEW_ELEMENT.
    * lo_te = view->GET_ELEMENT( id = 'MV' ).
    * view->REQUEST_FOCUS_ON_VIEW_ELEM( lo_te ).

  * request focus (7.02)
    view->REQUEST_FOCUS_ON_VIEW_ELEM( view->GET_ELEMENT( id = 'MV' ) ).

  * get the embedding window
    lr_window_controller = view->if_wd_view_controller-get_embedding_window_ctlr( ).
    lr_window_manager = lr_window_controller->get_window( ).

    lr_window_manager->subscribe_to_button_event( button = if_wd_window->co_button_ok
        action_name = 'ON_MV_OK'
        action_view = view
        is_default_button = abap_false ).

    lr_window_manager->SET_CLOSE_IN_ANY_CASE( abap_false ).

    wd_this->mo_window = lr_window_manager.

endmethod.
```
Note that nested method calls are not possible in SAP NetWeaver 7.0 and 7.01. Therefore, in this case, you must use the coding in the comment to request the focus on the TextEdit UI element.

4. Creating a Window to Embed the New View into

In this step, you create an additional window in which you embed the new view (MULTI_VALUE).

1. Select the Windows node in the component object list.
2. In the context menu, choose Create as Enhancement.
3. Enter a name for the new window (e.g. ENH_WND_MV).
4. In the Enhancement Implementations node, open the Windows folder and double-click on window ENH_WND_MV.
5. In the window structure, select the window ENH_WND_MV.
6. In the context menu, choose Embed View and select the view you have just created (MULTI_VALUE). The window structure now looks as follows:

   ![Window Structure Diagram]

5. Enhancing the EXTENDED_OPTIONS View

In the following steps, you enhance the EXTENDED_OPTIONS view to enable the navigation to the view you created in step 3.1. A hypertext link should appear on the right hand side of the ToolBar:

   ![Multiple Selection Diagram]

5.1 Inserting a ToolBarLinkToAction UI Element

You first enhance the EXTENDED_OPTIONS view by adding a new UI element.

1) Navigate to the EXTENDED_OPTIONS view.
2) In the UI element tree, select the ToolBar UI element.
3) In the context menu, choose Insert Right-Aligned Toolbar Element and from the dropdown list select ToolBarLinkToAction.

4) Enter a unique ID (e.g. LINK_MULTI_VALUE_PASTE).

5) For property text enter ‘Multi Value Paste’ or a similar text.

5.2 Creating an onAction Event Handler for the ToolBarLinkToAction

If the user navigates to the Multi Value Paste link, a dialog box should appear that allows the user to enter values. To achieve this behavior, you implement an onAction event handler for the ToolBarLinkToAction.

1. Below the Events header of the new ToolBarLinkToAction UI element, navigate to the OnAction row:

   ![Screenshot of the Event Handler for ToolBarLinkToAction]

2. In the value field, enter a name (e.g. MV_POPUP)

3. Double-click on the onAction row.

4. Use the Web Dynpro Code Wizard to generate the code for the onAction event handler method.

   In the Web Dynpro Statement Structure dialog box, select the Generate Popup checkbox. In the Window Name input field, enter the name of the window you created in step 4 (e.g. ENH_WND_MV):
5.3 Creating a Post-Exit in WDDOMODIFYVIEW of View EXTENDED_OPTIONS

The content of the table that is displayed in the Multiple Selection screen is created dynamically. As we will update the context later on, we must store a reference of the context node ‘CONTENT’ to the attribute created in step 7.

1. Open the Methods tab.
2. In the WDDOMODIFYVIEW row, choose the field in the Post-Exit column.
3. Create a post-exit for WDDOMODIFYVIEW. Use the string literal 'CONTENT' for method GET_CHILD_NODE:

```java
method PST_00O2TH202RTE8C2ZFFF7EEU4G "Exit of WDDOMODIFYVIEW (in ZZ_ENH_SEL_OPT )"

    wd_comp_controller->mo_context_node = wd_context->get_child_node( 'CONTENT' ).

endmethod.
```
6. (Optional) Influencing the Availability of the Multi-Value Paste Functionality

You may want to make your enhancement implementation available only under certain conditions. For example, you can bind it to the special privilege of the user. To do this, you can implement a post-exit for WDDOINIT.

1. In the Context tab of the EXTENDED_OPTIONS view, create a new context attribute (e.g. ENH_ACTIVE) of type WDY_BOOLEAN.
2. Bind the property visible of the ToolBarLinkToAction to this new context attribute.
3. In the EXTENDED_OPTIONS view, open the Methods tab.
4. Create a post-exit for WDDOINIT. You can use the following code as template:

```abap

if <something is valid>

    DATA lo_el_context TYPE REF TO if_wd_context_element.
    DATA ls_context TYPE wd_this->Element_context.

    * get element via lead selection
    lo_el_context = wd_context->get_element( ).

    * @TODO handle not set lead selection
    IF lo_el_context IS INITIAL.
    ENDIF.

    * set single attribute
    lo_el_context->set_attribute( name = 'ENH_ACTIVE' 
    value = abap_true ).

endif.

endmethod.
```
7. Testing your Enhancement Implementation

To test your enhancement implementation, you can use component WDR_TEST_SELECT_OPTIONS.
RELATED CONTENT

SAP Online Help - 7.02 Select Options 1.0
SAP Online Help - 7.02 Enhancement Implementations