



Ramakrishnappa Gangappa

March 10, 2014 14 minute read

Attach files to GOS with Save, Retrieve, Delete functionality in SAP Web Dynpro ABAP – Part 1

[Follow](#)

[RSS feed](#)

[Like](#)

9 Likes 25,144 Views 10 Comments

Purpose:

Application for attaching files to GOS with save, retrieve & delete functionality in Webdynpro ABAP

Scenario:

I would like to explain the functionality of how to save, retrieve & delete files / Notes during attachments to GOS in Webdynpro ABAP application.

Please refer the below link for more info on GOS

GOS (Generic Object Services)

Here I would consider the scenario of SAP FSRI module, and attach files / notes against a policy number.

Class: CL_GOS_API

Provides access to the GOS attachment list for applications. In the first step, the application must create an instance of the class CL_GOS_API and then register its unique application object at the same time. Currently, application objects of the categories BOR Object and Persistent Classes are supported. The attachment list can then be read, and the content of individual attachments can be read, modified, deleted, or created.

Pre-requisite:

Basic knowledge of [Webdynpro ABAP](#), & [OO ABAP](#)

Step by Step Process:

Step 1:

Go t-code SE11 and create a custom structure YST_GOS_ATTACH_LIST as below

i.e. add a field ICON and include structure GOS_S_ATTA as below

The screenshot shows the SAP Dictionary: Change Structure screen for the structure YST_GOS_ATTACH_LIST. The structure is active and has a short description of "Structure for Attachment list". The Components tab is selected, showing a table of components. The table has columns: Component, Typing Method, Component Type, Data Type, Length, Deci..., and Short Description. The components listed are ICON (CHAR, 30, 0 Icon name) and .INCLUDE (Types, GOS_S_ATTA, 0 GOS API: Structure for Attachment List). The components are highlighted with a red box.

Component	Typing Method	Component Type	Data Type	Length	Deci...	Short Description
ICON	Types	ICON_NAME	CHAR	30	0	Icon name
.INCLUDE	Types	GOS_S_ATTA		0	0	GOS API: Structure for Attachment List

Dictionary: Display Structure

| Hierarchy Display Append Structure...

Structure: Active
 Short Description:

Component	Typing Method	Component Type	Data Type	Length	Deci...	Short Description
ICON	Types	▼ ICON_NAME	CHAR	30		0 Icon name
.INCLUDE	Types	▼ GOS_S_ATTIA	□□	0		0 GOS API: Structure for Attachment List
.INCLUDE	Types	▼ GOS_S_ATTKEY	□□	0		0 GOS API: Key Structure of Attachment
ATTA_ID	Types	▼ GOS_ATTIA_ID	STRING	0		0 GOS API: Attachment ID
ATTA_CAT	Types	▼ GOS_ATTIA_CAT	CHAR	3		0 GOS API: Attachment Category
CR_USER	Types	▼ SO_CRO_NAM	CHAR	12		0 Creator Name
CR_NAME	Types	▼ SGS_CRNAME	CHAR	60		0 Creator Name
CR_DATE	Types	▼ SGS_CRDATE	DATS	8		0 Created On
CR_TIME	Types	▼ SGS_CRTIME	TIMS	6		0 Time created
CH_USER	Types	▼ SO_CHO_NAM	CHAR	12		0 Changed by
CH_NAME	Types	▼ SGS_CRNAME	CHAR	60		0 Creator Name
CH_DATE	Types	▼ SGS_CRDATE	DATS	8		0 Created On
CH_TIME	Types	▼ SGS_CRTIME	TIMS	6		0 Time created
FILESIZE	Types	▼	INT4	10		0 File Size
FILENAME	Types	▼ BITM_FILENAME	CHAR	255		0 File Name of Browser Item (Display Attribute)
TECH_TYPE	Types	▼ BITM_TYPE	CHAR	40		0 Type Information for Browser Item
DESCR	Types	▼ BITM_DESCR	CHAR	60		0 Description for Browser Item (Display Attribute)
LANG	Types	▼ SYLANGU	LANG	1		0 Language Key of Current Text Environment

Step 2:

Go to t-code **SE80** and create a web dynpro component as below

Web Dynpro: Component / Create Interface



Name

Description

Type ☒ Web Dynpro Component
☐ Web Dynpro Component Interface

Window Name

View Name



 

Step 3:


Go to context tab of component controller and create a node ATTACHMENTS by using structure YST_GOS_ATTACH_LIST and set the properties as below


Component Controller COMPONENTCONTROLLER Active(revised)

















Properties Context Attributes Events Methods

Controller Usage  

Context COMPONENTCONTROLLER

▼  CONTEXT

▼  ATTACHMENTS

-  ICON
-  ATTA_ID
-  ATTA_CAT
-  CR_USER
-  CR_NAME
-  CR_DATE
-  CR_TIME
-  CH_USER
-  CH_NAME
-  CH_DATE
-  CH_TIME
-  FILESIZE
-  FILENAME
-  TECH_TYPE
-  DESCR
-  LANG

Property	Value
<u>Nodes</u>	
Node Name	ATTACHMENTS
Interface Node	<input type="checkbox"/>
Input Element (Ext.)	<input type="checkbox"/>
Dictionary structure	YST_GOS_ATTACH_LIST
Cardinality	0..n
Selection	0..1
Initialization Lead Selection	<input type="checkbox"/>
Singleton	<input type="checkbox"/>
Supply Function	

Step 4:

Create another node "ATTACHMENT_CONTENTS" by using structure GOS_S_ATTCONT as below

Component Controller

COMPONENTCONTROLLERActive(revised)

PropertiesContextAttributesEventsMethods

Controller Usage

Context COMPONENTCONTROLLER

CONTEXT

ATTACHMENTS

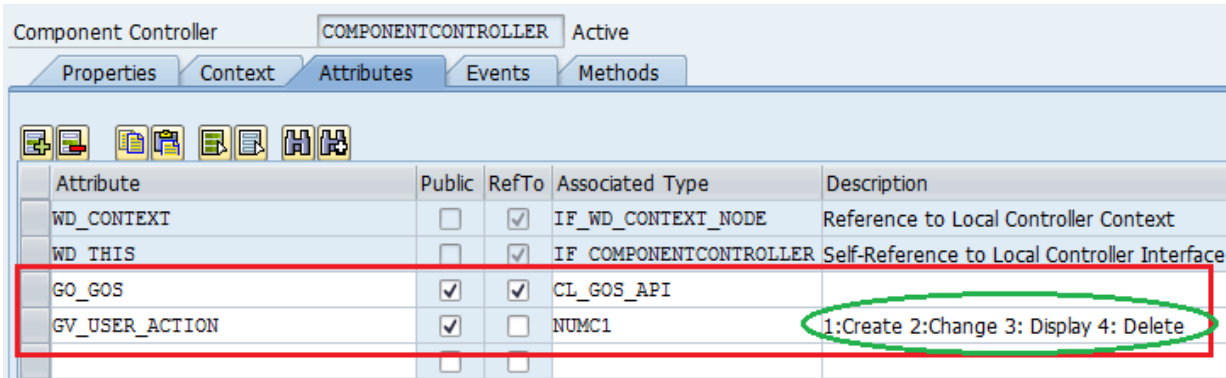
ATTACHMENT_CONTENTS

ATTA_ID
ATTA_CAT
CR_USER
CR_NAME
CR_DATE
CR_TIME
CH_USER
CH_NAME
CH_DATE
CH_TIME
FILESIZE
FILENAME
TECH_TYPE
DESCR
LANG
CONTENT
CONTENT_X

Property	Value
Nodes	
Node Name	ATTACHMENT_CONTENTS
Interface Node	<input type="checkbox"/>
Input Element (Ext.)	<input type="checkbox"/>
Dictionary structure	GOS_S_ATTCONT
Cardinality	1..1
Selection	0..1
Initialization Lead Selection	<input checked="" type="checkbox"/>
Singleton	<input type="checkbox"/>
Supply Function	

Step 5:

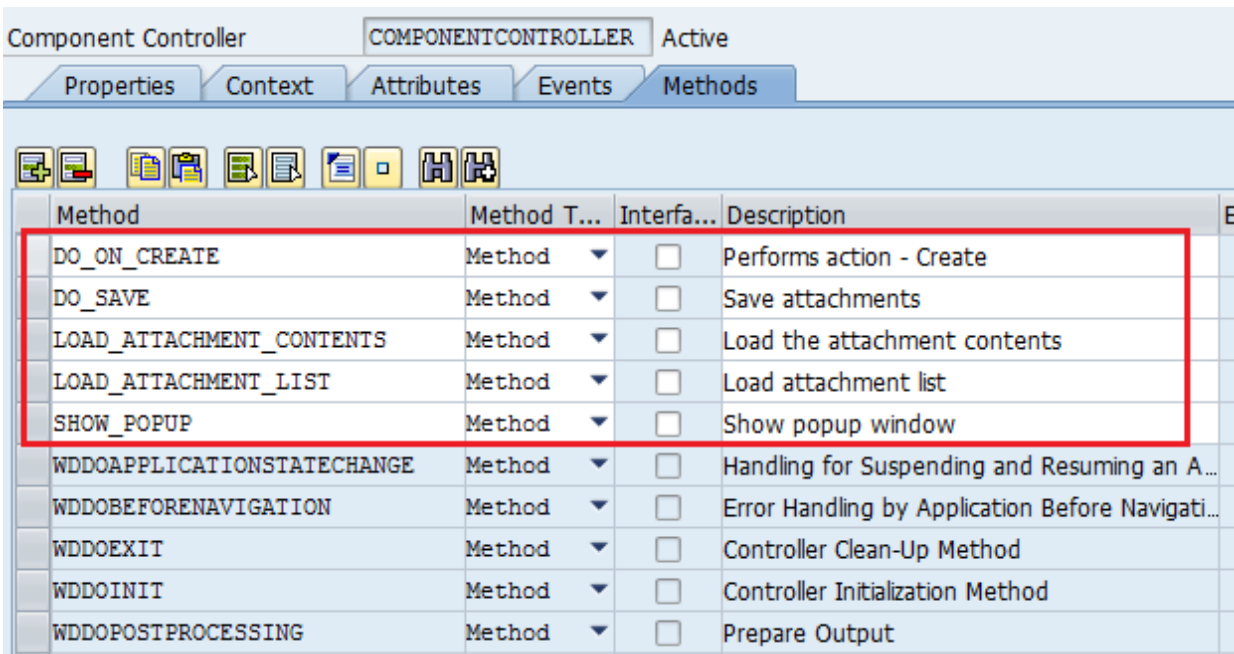
Go to attributes tab of component controller and create the global attributes GO_GOS & GV_USER_ACTION as below



Attribute	Public	RefTo	Associated Type	Description
WD_CONTEXT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	IF_WD_CONTEXT_NODE	Reference to Local Controller Context
WD_THIS	<input type="checkbox"/>	<input checked="" type="checkbox"/>	IF_COMPONENTCONTROLLER	Self-Reference to Local Controller Interface
GO_GOS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CL_GOS_API	
GV_USER_ACTION	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NUMC1	1:Create 2:Change 3: Display 4: Delete

Step 5A:

Create methods in component controller as below



Method	Method T...	Interfa...	Description
DO_ON_CREATE	Method ▼	<input type="checkbox"/>	Performs action - Create
DO_SAVE	Method ▼	<input type="checkbox"/>	Save attachments
LOAD_ATTACHMENT_CONTENTS	Method ▼	<input type="checkbox"/>	Load the attachment contents
LOAD_ATTACHMENT_LIST	Method ▼	<input type="checkbox"/>	Load attachment list
SHOW_POPUP	Method ▼	<input type="checkbox"/>	Show popup window
WDDOAPPLICATIONSTATECHANGE	Method ▼	<input type="checkbox"/>	Handling for Suspending and Resuming an A...
WDDOBEFORENAVIGATION	Method ▼	<input type="checkbox"/>	Error Handling by Application Before Navigati...
WDDOEXIT	Method ▼	<input type="checkbox"/>	Controller Clean-Up Method
WDDOINIT	Method ▼	<input type="checkbox"/>	Controller Initialization Method
WDDOPOSTPROCESSING	Method ▼	<input type="checkbox"/>	Prepare Output

Now, the write the code as below

Method: DO_ON_CREATE

Method DO_ON_CREATE				
<div><div><div><div></div><div></div><div></div><div></div><div></div></div><div>Exceptions</div></div></div>				
Parameter	Type	RefTo	Opt	Associated Type
ID_WINDOW_NAME	Impor...	<input type="checkbox"/>	<input type="checkbox"/>	STRING
IO_VIEW_CTRL	Impor...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	IF_WD_VIEW_CONTROLLER

DO_ON_CREATE

DO_ON_CREATE	
METHOD do_on_create . " Set user action "=====	
wd_this->gv_user_action = 1."Create	
 " Refresh attachment list ctx "=====	
DATA lo_node TYPE REF TO if_wd_context_node.	
 lo_node = wd_context->get_child_node(name = wd_this->wdctx_attachment_contents).	
 lo_node->invalidate().	
 " Show respective popup window "=====	
wd_this->show_popup(EXPORTING io_view_ctrl = io_view_ctrl id_button_kind = 3 id_action1 = 'SAVE' id_window_name = id_window_name).	
ENDMETHOD.	

Method: DO_SAVE

DO_SAVE	
METHOD do_save .	

DO_SAVE

```
DATA: ls_attacont TYPE gos_s_attcont,  
      ls_atta_key TYPE gos_s_attkey,  
      lv_rctype   TYPE oblrcotype,  
      lv_commit   TYPE flag,  
      lo_node     TYPE REF TO if_wd_context_node,  
      lo_element  TYPE REF TO if_wd_context_element.
```

" Read notes detail context

"=====

```
lo_node = wd_context->get_child_node(  
    name    = wd_this->wdctx_attachment_contents  
).  
  
lo_node->get_element( )->get_static_attributes(  
    IMPORTING static_attributes = ls_attacont ).  
"
```

" Attach notes

"=====

TRY.

```
ls_attacont-atta_cat = cl_gos_api=>c_msg.
```

" Check if attachment documents

IF ls_attacont-content_x IS INITIAL.

```
ls_attacont-tech_type = 'TXT'.
```

```
lv_rctype = cl_gos_api=>c_annotation.
```

ELSE.

```
lv_rctype = cl_gos_api=>c_attachment.
```

ENDIF.

IF wd_this->gv_user_action EQ 1."Create

```
wd_this->go_gos->insert_al_item(  
    EXPORTING
```

```
is_attcont = ls_attacont
```

DO_SAVE

```
    iv_roltype = lv_roltype
RECEIVING
    rv_commit = lv_commit
).

ELSEIF wd_this->gv_user_action EQ 2."Change
    lv_commit =
        wd_this->go_gos->update_al_item( is_attcont = ls_attacont ).
ELSE."delete

    MOVE-CORRESPONDING ls_attacont TO ls_atta_key.

    lv_commit =
        wd_this->go_gos->delete_al_item( is_atta_key = ls_atta_key ).
*    CATCH cx_gos_api. " GOS API: Exception
ENDIF.

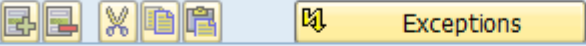
    CATCH cx_gos_api.                "#EC NO_HANDLER
ENDTRY.

IF lv_commit EQ abap_true.
    CALL FUNCTION 'BAPI_TRANSACTION_COMMIT'
        EXPORTING
            wait = abap_true.
ENDIF.

"=====
" Reload the context
"=====

    wd_this->load_attachment_list( ).
ENDMETHOD.
```

Method: LOAD_ATTACHMENT_CONTENTS

Method				
LOAD_ATTACHMENT_CONTENTS				
				
Parameter	Type	RefTo	Opt	Associated Type
IO_VIEW_CTRL	Impor...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	IF_WD_VIEW_CONTROLLER
IS_ATT	Impor...	<input type="checkbox"/>	<input type="checkbox"/>	YST_GOS_ATTACH_LIST

LOAD_ATTACHMENT_CONTENTS	
METHOD load_attachment_contents . DATA: ls_atta_key TYPE gos_s_attkey, ls_attacont TYPE gos_s_attcont, lv_window_name TYPE string, lv_title TYPE string, lv_text TYPE string, lv_button_kind TYPE wdr_popup_button_kind. " Prepare key MOVE-CORRESPONDING is_atta TO ls_atta_key. "get attachment content TRY. ls_attacont = wd_this->go_gos->get_al_item(is_atta_key = ls_atta_key). CATCH cx_gos_api. " #EC NO_HANDLER ENDTRY. "set data to context DATA lo_node TYPE REF TO if_wd_context_node. DATA lo_element TYPE REF TO if_wd_context_element. lo_node = wd_context->get_child_node(name = wd_this->wdctx_attachment_contents).	

LOAD_ATTACHMENT_CONTENTS

```
lo_element = lo_node->get_element( ).

lo_element->set_static_attributes( static_attributes = ls_attacont ).

" Based on content, show popup window
"=====
CLEAR lv_button_kind.
IF wd_this->gv_user_action EQ 2.
  lv_title = 'Change'.
  lv_button_kind = 3.
ELSEIF wd_this->gv_user_action EQ 3.
  lv_title = 'Display'.
ELSEIF wd_this->gv_user_action EQ 4.
  lv_title = 'Delete'.
  lv_button_kind = 3.
ENDIF.

IF ls_attacont-content_x IS INITIAL.
  lv_window_name = 'W_ATTACH_NOTES'.
  lv_text = 'Notes'.
  CONCATENATE lv_title lv_text INTO lv_title SEPARATED BY space.
ELSE.
  lv_window_name = 'W_ATTACH_FILES'.
  lv_text = 'File'.
  CONCATENATE lv_title lv_text INTO lv_title SEPARATED BY space.

ENDIF.

wd_this->show_popup(
  EXPORTING
    io_view_ctrl  = io_view_ctrl
    id_button_kind = lv_button_kind
    id_title      = lv_title
```

LOAD_ATTACHMENT_CONTENTS
<pre> id_action1 = 'SAVE' id_window_name = lv_window_name). ENDMETHOD.</pre>

Method: LOAD_ATTACHMENT_LIST

LOAD_ATTACHMENT_LIST
<pre> METHOD load_attachment_list . "===== DATA: lt_atta TYPE TABLE OF gos_s_atta, lt_attachment_list TYPE wd_this->elements_attachments, ls_attachment_list LIKE LINE OF lt_attachment_list. FIELD-SYMBOLS: <ls_atta> LIKE LINE OF lt_atta. TRY. lt_atta = wd_this->go_gos->get_atta_list(). CATCH cx_gos_api. "#EC NO_HANDLER ENDTRY. CLEAR lt_attachment_list. LOOP AT lt_atta ASSIGNING <ls_atta>. CLEAR ls_attachment_list. MOVE-CORRESPONDING <ls_atta> TO ls_attachment_list. CONDENSE ls_attachment_list-tech_type. IF ls_attachment_list-tech_type EQ 'TXT' OR ls_attachment_list-tech_type IS INITIAL.</pre>

LOAD_ATTACHMENT_LIST

```
ls_attachment_list-icon = 'ICON_ANNOTATION'.
ELSE.
ls_attachment_list-icon = 'ICON_ATTACHMENT'.
ENDIF.
APPEND ls_attachment_list TO lt_attachment_list.
ENDLOOP.

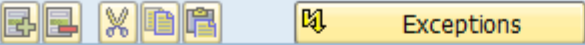
"=====
" Set data to context node
"=====
DATA lo_node TYPE REF TO if_wd_context_node.

lo_node = wd_context->get_child_node(
    name    = wd_this->wdctx_attachments
).

lo_node->bind_table(
    EXPORTING
        new_items    = lt_attachment_list
).

ENDMETHOD.
```

Method: SHOW_POPUP

Method SHOW_POPUP					
					
Parameter	Type	RefTo	Opt	Associated Type	Default Value
ID_ACTION1	Impor...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	STRING	
ID_ACTION1_TEXT	Impor...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	STRING	'Ok'
ID_ACTION2	Impor...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	STRING	
ID_BUTTON_KIND	Impor...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	WDR_POPUP_BUTTON_KIND	2
ID_RESIZABLE	Impor...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	WDY_BOOLEAN	abap_false
ID_TITLE	Impor...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	STRING	
ID_WINDOW_NAME	Impor...	<input type="checkbox"/>	<input type="checkbox"/>	STRING	
IO_VIEW_CTRL	Impor...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	IF_WD_VIEW_CONTROLLER	

SHOW_POPUP	
<pre> method SHOW_POPUP . "Purpose: Display popup window of More CoB "----- "Data declarations "----- DATA lo_window_manager TYPE REF TO if_wd_window_manager. DATA lo_api_component TYPE REF TO if_wd_component. DATA lo_window TYPE REF TO if_wd_window. DATA lv_title TYPE string. "----- "get view controller, window manager lo_api_component = wd_this->wd_get_api(). lo_window_manager = lo_api_component->get_window_manager(). * "----- lv_title = id_title. IF lv_title IS INITIAL. lv_title = 'Create Attachments'. ENDIF. </pre>	

SHOW_POPUP

“creat a pop up window

CALL METHOD lo_window_manager->create_window

EXPORTING

```
* modal          = ABAP_TRUE
window_name      = id_window_name
title            = lv_title
* close_button   = ABAP_TRUE
button_kind      = id_button_kind
* message_type   = IF_WD_WINDOW=>CO_MSG_TYPE_NONE
* close_in_any_case = ABAP_TRUE
* message_display_mode =
* default_button =
is_resizable     = abap_false
```

RECEIVING

```
window          = lo_window.
```

IF lo_window IS BOUND.

IF id_button_kind EQ 3.

IF id_action1 IS NOT INITIAL.

CALL METHOD lo_window->subscribe_to_button_event

EXPORTING

```
button          = 4
button_text      = id_action1_text
* tooltip        =
action_name      = id_action1
action_view      = io_view_ctrl
is_default_button = abap_true.
```

ENDIF.

IF id_action2 IS NOT INITIAL.

CALL METHOD lo_window->subscribe_to_button_event

EXPORTING



SHOW_POPUP	
	<pre> button = 6 * button_text = * tooltip = action_name = id_action2 action_view = io_view_ctrl is_default_button = abap_false. ENDIF. ENDIF. lo_window->open(). ENDIF. "----- "----- endmethod.</pre>

Step 6:

Go to view V_MAIN and create context attribute POLICY_ID under node CONTEXT as below

View V_MAIN Active(revised)

Properties Layout Inbound Plugs Outbound Plugs Context Attributes Actions Methods

Controller Usage  

Context V_MAIN

CONTEXT

POLICY_ID

YDEMO_ATTACH_FILES_GOS.COMPONENTCONTROLL

Context COMPONENTCONTROLLER

CONTEXT

ATTACHMENTS

ATTACHMENT_CONTENTS

Property	Value
Attribute	
Attribute Name	POLICY_ID
Type assignment	Type
Type	/MSG/H_POL_ID
Read-only	<input type="checkbox"/>
Default Value	
Null Value	<input type="checkbox"/>
Input Help Mode	Automatic
Determined Input Help	/MSG/H_POL
Type of Input Help	Search help
Formatting	
Compression	Default Value

Step 7:

Map the context node ATTACHMENTS from component controller to V_MAIN view's controller as below

View V_MAIN Active

Properties Layout Inbound Plugs Outbound Plugs Context Attributes Actions Methods

Controller Usage

Context V_MAIN

- CONTEXT
 - ATTACHMENTS
 - POLICY_ID

Drag & Drop

Context COMPONENTCONTROLLER

- CONTEXT
 - ATTACHMENTS
 - ATTACHMENT_CONTENTS

Property	Value
Nodes	
Node Name	ATTACHMENTS
Dictionary structure	YST_GOS_ATTACH_LIST
Cardinality	0..n
Selection	0..1
Initialization Lead Selection	<input type="checkbox"/>
Singleton	<input type="checkbox"/>
Supply Function	
Mapping Path	YDEMO_ATTACH_FILES_GOS.COMPONENTCONTROLLER.ATTACHMENTS

Step 8:

Create an input field and bind the value to context attribute POLICY_ID as below

V_MAIN Inactive(revised)

Inbound Plugs Outbound Plugs Context Attributes Actions Methods

V_MAIN.POLICY_ID

- ☐ CONTEXT_MENUS
- ☒ ROOTUIELEMENTCONTAINER
 - ☒ POLICY ID LBL
 - ☒ POLICY ID

Property	Value	Binding
Properties (InputField)		
ID	POLICY_ID	
activateAccessKey	<input type="checkbox"/>	
alignment	auto	
contextMenuBehaviour	Inherit	
contextMenuId		
datePickerReferenceId		
displayAsText	<input type="checkbox"/>	
displayOnly	<input type="checkbox"/>	
enabled	<input checked="" type="checkbox"/>	
explanation		
imeMode	auto	
inputPrompt		
length	0	
noHistory	<input type="checkbox"/>	
passwordField	<input type="checkbox"/>	
readOnly	<input type="checkbox"/>	
state	Normal Item	
suggestFilterMethod	prefixSearch	
suggestValues	<input type="checkbox"/>	
suppressValueHelp	<input type="checkbox"/>	
textDirection	Inherit	
tooltip		
value	V_MAIN.POLICY_ID	
visible	Visible	
width		

Step 9:

Create a button "BTN_GET_LIST" and create an action "GET_LIST" and bind it to the event of button as below

- ☐ CONTEXT_MENUS
- ▼ ☐ ROOTUIELEMENTCONTAINER
 - **T** POLICY_ID_LBL
 - ☐ POLICY_ID
 - ☒ **BTN_GET_LIST**

Property	Value	Binding
Properties (Button)		
ID	BTN_GET_LIST	
activateAccessKey	<input type="checkbox"/>	
contextMenuBehaviour	Inherit	
contextMenuId		
design	next	
enabled	<input checked="" type="checkbox"/>	
explanation		
hotkey	none	
imageFirst	<input checked="" type="checkbox"/>	
imageIsDecorative	<input type="checkbox"/>	
imageSource		
text	Get Attachment List	
textDirection	Inherit	
tooltip		
visible	Visible	
width		
Events		
onAction	GET_LIST	
Layout Data (FlowData)		
cellDesign	padless	
vGutter	None	

Step 9 A:

Write the below code in event handler method ONACTIONGET_LIST as below

Here, we are creating an instance to GOS for the given policy number.

ONACTIONGET_LIST

METHOD onactionget_list .

” Data declarations

DATA lv_pol_id TYPE /msg/h_pol_id.

“=====

“read policy number

“=====

wd_context->get_attribute(

EXPORTING

name = 'POLICY_ID'

IMPORTING

value = lv_pol_id

).

“=====

“Create an instance of GOS

“=====

DATA ls_object TYPE gos_s_obj.

CLEAR ls_object.

ls_object-instid = lv_pol_id. ” Policy Number

ls_object-typeid = '/MSG/HRISK'.

ls_object-catid = 'BO'.

TRY.

wd_comp_controller->go_gos =

cl_gos_api=>create_instance(is_object = ls_object).

CATCH cx_gos_api. “#EC NO_HANDLER

ENDTRY.

“=====

“Load attachment list








“=====

ONACTIONGET_LIST
wd_comp_controller->load_attachment_list(). ENDMETHOD.

Step 10:

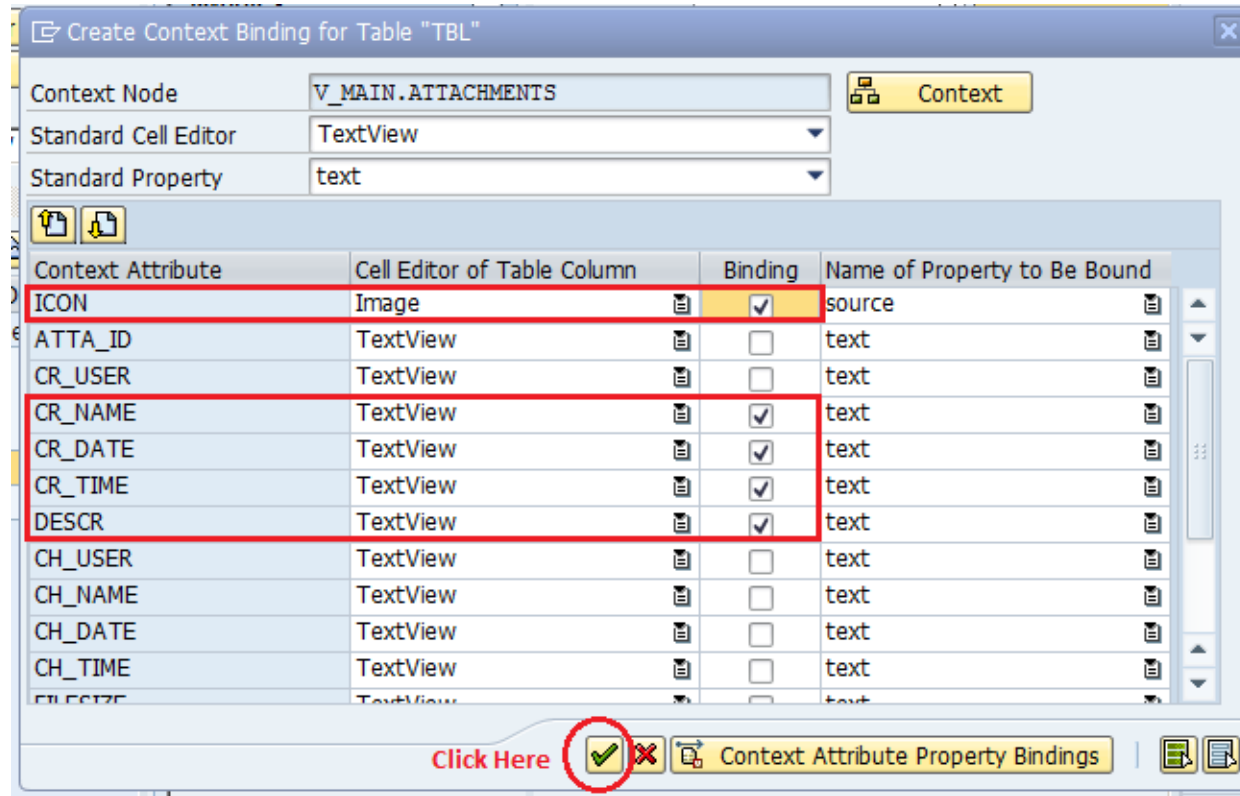
Create a table ui element TBL and bind the data source property to node ATTACHMENTS as below

- ☐ CONTEXT_MENUS
- ▼ ☐ ROOTUIELEMENTCONTAINER
 - ▶ ☐ TC_TOP
 - ▼ ☐ TC
 - ▶ ☒ TBL

Property	Value
<u>Properties (Table)</u>	
ID	TBL
accessibilityDescription	
activateAccessKey	<input type="checkbox"/>
columnResizeMode	preserveTableWidth 
contextMenuBehaviour	Inherit 
contextMenuId	
dataSource	V_MAIN.ATTACHMENTS
defaultButtonId	
design	Standard 
displayEmptyRows	<input type="checkbox"/>
dropOnRowName	
emptyTableText	
enabled	<input checked="" type="checkbox"/>
firstActualRow	0
firstVisibleRow	0
firstVisibleScrollableCol	
fixedTableLayout	<input type="checkbox"/>
footerVisible	<input checked="" type="checkbox"/>
gridMode	both 
handleHotkeys	<input type="checkbox"/>
legendId	
multiColSorting	<input type="checkbox"/>
readOnly	<input type="checkbox"/>
rowCount	-1
rowSelectable	<input checked="" type="checkbox"/>
scrollableColCount	-1
selectedPopin	
selectionChangeBehavio...	auto 
selectionColumnDesign	Left 

Step 11:

Create the columns and bind to the context attributes as shown below



Step 12:

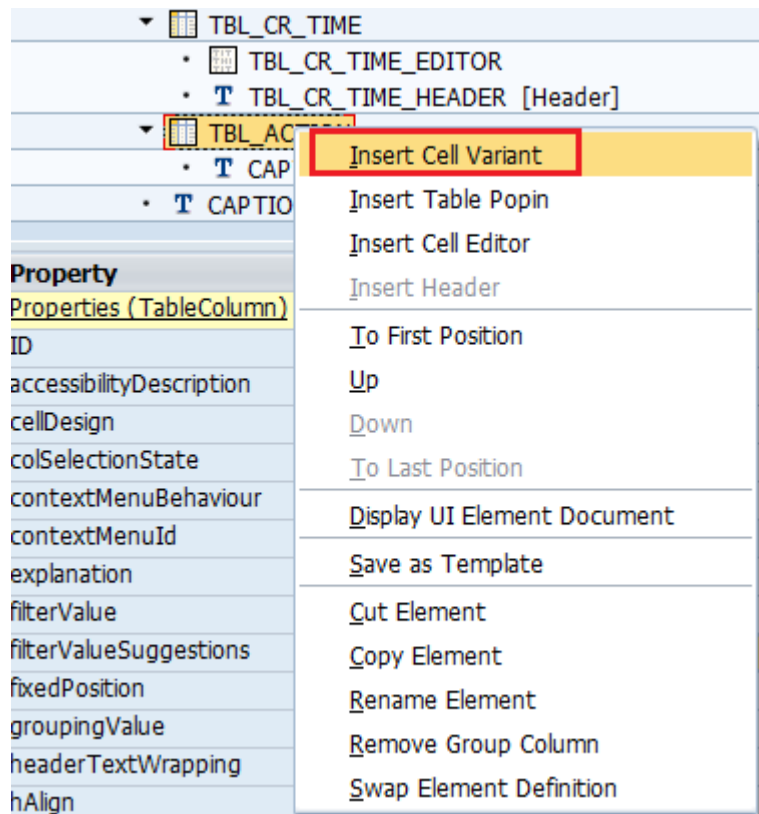
Create an extra column called "ACTIONS" for display, change, delete actions with SelectedCellVariant "KEY" as below

- ▶ TBL_CR_NAME
- ▶ TBL_CR_DATE
- ▶ TBL_CR_TIME
- ▶ **TBL_ACTION**
- CAPTION [Header]
- ▶ TOOLBAR [Toolbar]

Property	Value	Binding
Properties (TableColumn)		
ID	TBL_ACTION	
accessibilityDescription		
cellDesign	standard	
colSelectionMode	notSelectable	
contextMenuBehaviour	Inherit	
contextMenuId		
explanation		
filterValue		
filterValueSuggestions		
fixedPosition	notFixed	
groupingValue		
headerTextWrapping	<input type="checkbox"/>	
hAlign	auto	
isFiltered	<input type="checkbox"/>	
resizable	<input checked="" type="checkbox"/>	
selectedCellVariant	KEY	
sortState	None	
visible	Visible	
width		
Events		
onAction		

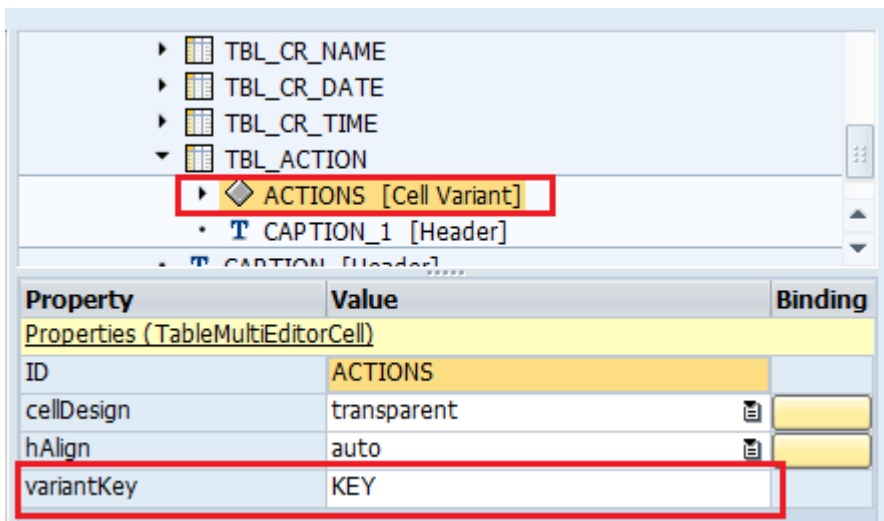
Step 13:

Right click on column ACTIONS and choose "Insert Cell Variant" as below



Step 14:

Set the variantKey "KEY" to TableMultiCellEditor as shown below



Step 15 :

Create a button for display and bind it to the action DO_ACTION as below

TBL_CR_TIME

TBL_CR_TIME_EDITOR

TBL_CR_TIME_HEADER [Header]

TBL_ACTION

ACTIONS [Cell Variant]

BTN_DISPLAY

CARTON 1 [Header]

Property	Value	Binding
Properties (Button)		
ID	BTN_DISPLAY	
activateAccessKey	<input type="checkbox"/>	
contextMenuBehaviour	Inherit	
contextMenuId		
design	standard	
enabled	<input checked="" type="checkbox"/>	
explanation		
hotkey	none	
imageFirst	<input checked="" type="checkbox"/>	
imageIsDecorative	<input type="checkbox"/>	
imageSource	~Icon/Display	
text		
textDirection	Inherit	
tooltip		
visible	Visible	
width		
Events		
onAction	DO_ACTION	

Step 16 :

Create a button for change and bind it to the action DO_ACTION as below

- TBL_CR_TIME_HEADER [Header]
- TBL_ACTION
 - ACTIONS [Cell Variant]
 - BTN_DISPLAY
 - **BTN_CHANGE**
 - CAPTION_1 [Header]

Property	Value	Binding
Properties (Button)		
ID	BTN_CHANGE	
activateAccessKey	<input type="checkbox"/>	
contextMenuBehaviour	Inherit	
contextMenuId		
design	standard	
enabled	<input checked="" type="checkbox"/>	
explanation		
hotkey	none	
imageFirst	<input checked="" type="checkbox"/>	
imageIsDecorative	<input type="checkbox"/>	
imageSource	~Icon/Edit	
text		
textDirection	Inherit	
tooltip		
visible	Visible	
width		
Events		
onAction	DO_ACTION	

Step 17 :

Create a button for delete and bind it to the action DO_ACTION as below

ACTIONS [Cell Variant]

- BTN_DISPLAY
- BTN_CHANGE
- BTN_DELETE**
- CAPTION_1 [Header]
- CAPTION [Header]
- TOOLBAR [Toolbar]

Property	Value	Bindi...
Properties (Button)		
ID	BTN_DELETE	
activateAccessKey	<input type="checkbox"/>	
contextMenuBehaviour	Inherit	
contextMenuId		
design	standard	
enabled	<input checked="" type="checkbox"/>	
explanation		
hotkey	none	
imageFirst	<input checked="" type="checkbox"/>	
imageIsDecorative	<input type="checkbox"/>	
imageSource	~Icon/Delete	
text		
textDirection	Inherit	
tooltip		
visible	Visible	
width		
Events		
onAction	DO_ACTION	

Step 17A:

Write the below code in event handler method ONACTIONDO_ACTION

Event Handler

ONACTIONDO_ACTION

Parameter	RefTo	Opt	Associated Type	Description
WDEVENT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	CL_WD_CUSTOM_EVENT	
CONTEXT_ELEMENT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	IF_WD_CONTEXT_ELEMENT	
ID	<input type="checkbox"/>	<input type="checkbox"/>	STRING	

ONACTIONDO_ACTION

ONACTIONDO_ACTION

METHOD onactiondo_action .

DATA ls_attachments TYPE wd_this->element_attachments.

DATA lo_view_ctrl TYPE REF TO if_wd_view_controller.

CASE id.

WHEN 'BTN_CHANGE':

wd_comp_controller->gv_user_action = 2."Change

WHEN 'BTN_DISPLAY':

wd_comp_controller->gv_user_action = 3."Display

WHEN 'BTN_DELETE':

wd_comp_controller->gv_user_action = 4."Delete

WHEN OTHERS.

ENDCASE.

"get the current row data

CALL METHOD context_element->get_static_attributes

IMPORTING

static_attributes = ls_attachments.

"=====

"Load attachment content

"=====

lo_view_ctrl ?= wd_this->wd_get_api().

CALL METHOD wd_comp_controller->load_attachment_contents

EXPORTING

is_atta = ls_attachments

io_view_ctrl = lo_view_ctrl.

ENDMETHOD.

Step 17B:

Go to actions tab of view V_MAIN and create an action SAVE for save action on popup window for attaching notes or files. i.e. we have subscribed the button “OK” of popup window to SAVE action on view V_MAIN

Write the below code in event handler method ONACTIONSAVE.

ONACTIONSAVE	
" Save attachements	
wd_comp_controller->do_save().	

Continued.....

Attach files to GOS with Save, Retrieve, Delete functionality in SAP Web Dynpro ABAP – Part 2

Alert Moderator

Assigned tags

UI Web Dynpro ABAP | abap web dynpro | abap webdynpo | attach files gos web dynpro abap | Web Dynpro |

[View more...](#)

Related Blog Posts

[Attach files to GOS with Save, Retrieve, Delete functionality in SAP Web Dynpro ABAP – Part 2](#)

By **Ramakrishnappa Gangappa** , Mar 12, 2014

[Attach files with save, retrieve and delete functionality in Web Dynpro ABAP – Part 1](#)

By **Ramakrishnappa Gangappa** , Jan 10, 2014

[Attach files with save, retrieve and delete functionality in Web Dynpro ABAP – Part 2](#)

By **Ramakrishnappa Gangappa** , Jan 10, 2014

Related Questions

[Display file in the desktop](#)

By **Former Member** , Aug 18, 2015

[Generic Object Services using WebDynpro ABAP](#)

By **Former Member** , Nov 02, 2008

[Display PO Attachment](#)

By **Former Member** , Mar 02, 2009

10 Comments

You must be [Logged on](#) to comment or reply to a post.



Former Member

[March 17, 2014 at 3:24 pm](#)

Hello

Can you add the implementation code for method LOAD_ATTACHMENT_LIST ?

And also code to implement for events DO_ACTION and GET_LIST ?

Thx

Xavier

Like (0)



[Ramakrishnappa Gangappa](#) | Post author

[March 18, 2014 at 1:54 am](#)

Hi Xavier,

Thanks a lot for going through the document in details 😊 ... I appreciate your findings.. yah! missed to include the implementation of those methods 😞 ..

I have added the implementation code of those methods 😊

Regards,

Rama

Like (0)



Former Member

[August 26, 2014 at 6:35 pm](#)

We need to find out the minimum Netweaver Enhancement Pack level that is required for CL_GOS_API – so we can get this to work in our webdynpro development.

Like (0)



[Ramakrishnappa Gangappa](#) | Post author

[August 27, 2014 at 3:45 am](#)

Hi,

As for as I know, the class CL_GOS_API is available from NW 7.31.

Like (0)



Former Member

[August 27, 2014 at 3:16 am](#)

Hi,

Class CL_GOS_API and structure GOS_S_ATTA does not exist in our system ECC 6.0,.

What to do? do we need to activate object or patch update?

Like (0)



[Ramakrishnappa Gangappa](#) | Post author

[August 27, 2014 at 3:47 am](#)

Hi John,

AFAIK,the class CL_GOS_API is available from NW 7.31. I think you might need to upgrade your system to EHP6.

Like (0)



Former Member

January 19, 2015 at 6:52 am

Dear Ramakrishnappa Gangappa,

Thanks for u r Sharing the Good Document regarding GOS In WebDynpro. I'm Getting the all that u r shared but after upload then i want to see the attachments like(PDF,PNG,JPEG,doc..) files are converted as a raw string data type data, I'm unable to get the exact output that i have uploaded. while click on download button it's path changed like C:/FAKEPATH/KOLA.jpg. Please help me regarding this.

Thanks & Regards,

Suresh Reddy.

Like (0)



Ramakrishnappa Gangappa | Post author

January 19, 2015 at 8:24 am

Hi Suresh,

Thank you for your feedback.

Please have look at the below thread:

[File path defaulted to 'C:\fakepath' when attaching](#)

Hope this helps you.

Regards,

Rama

Like (0)



Former Member

January 21, 2015 at 6:05 am

Dear Ramakrishnappa,

Thanks for your feedback i'm getting the exact output in internet Explorer Browser with your Suggestion but in the google chrome browser and also it's work for .PDF Files. Suppose if i upload the file formats like .PNG,.DOC,.JPEG and Excel then it woks like Pdf file viewing on the Browser. If is not possible what is alternative solution for this.

Thanks & Regards,

Suresh Reddy.

Like (0)



Former Member

October 18, 2016 at 5:27 pm

Hi

I am trying to do the same, But i am getting the Error:

500 SAP Internal Server Error

ERROR: Could not find attribute ATTACHMENTS (termination: RABAX_STATE)

please help me, I am new for WD.

Like (0)

Find us on

Privacy	Terms of Use
Legal Disclosure	Copyright
Trademark	Cookie Preferences
Newsletter	Support