

ABAP Control Framework

Contributed by Kevin Wilson
Tuesday, 20 May 2008

Custom Controls Tutorial Creating a Control STEP 1 - Create the Instance Define a reference variable for the Custom Container in which you want to place the custom control DATA container TYPE REF TO cl_gui_custom_container. Define a reference variable for the control object you wish to use DATA picture TYPE REF TO cl_gui_picture. Create the Custom Container. You must already have created the area 'CUSTOM_CONTAINER' for the Custom Container in the Screen Painter. CREATE OBJECT container

EXPORTING

container_name = 'CUSTOM_CONTAINER'

lifetime = lifetime.

STEP 2 - Register Events The event mechanism of the Control Framework allows you to use handler methods in your programs to react to events triggered by the control (for example, a double-click). Registering the events with the Control Framework DATA my_control TYPE REF TO cl_gui_xyz. DATA events TYPE cntl_simple_events.

DATA wa_events TYPE cntl_simple_event. wa_events-eventid = event_id .

wa_events-appl_event = appl_event .

APPEND wa_events TO events. CALL METHOD my_control->set_registered_events

events = events. Defining a handler method CLASS lcl_event_receiver DEFINITION.

PUBLIC SECTION.

METHODS Event_Handler

FOR EVENT event_name OF cl_gui_xyz

IMPORTING event_parameter

sender .

ENDCLASS. Registering the handler method DATA event_receiver TYPE REF TO lcl_event_receiver.

CREATE OBJECT event_receiver.

SET HANDLER event_receiver->Event_Handler

FOR my_control. Processing the event CLASS lcl_event_receiver IMPLEMENTATION.

METHOD Event_Handler.

* Event processing

ENDMETHOD

ENDCLASS. STEP 3 - Use the control STEP 4 - Destroy the control Use the method free to destroy the

Custom Control at the frontend. If you no longer need the control container, release it as well: CALL METHOD picture->free

EXCEPTIONS cntl_error = 1

cntl_system_error = 2. CALL METHOD container->free

EXCEPTIONS cntl_error = 1

cntl_system_error = 2

Pay careful attention to the sequence in which you destroy controls at the frontend

When you destroy a container, all controls in it are automatically destroyed as well. If you have already destroyed a control and try to destroy it again, an error occurs. You can check whether a control has already been

destroyed using the method is_alive. Delete the reference variables to the custom control and the control

container. FREE PICTURE. FREE CONTAINER. Event Handling See STEP 2 above

Drag and Drop Drag and drop allows the user to select an object from one part of a custom control (source) and drop it on another part of a custom control (target). An action occurs in the second part that depends on the object type. Source and target may be either the

same control or different controls. A particular drag and drop behavior is set for each custom control. This behavior may

be set globally for all elements of the control (for example, SAP Textedit), or you may be able to define a different

behavior for each component (for example SAP Tree). Each behavior consists of one or more descriptions. A

description has the following attributes: o DragSrc: Object is the source of a drag and drop procedure o DropTarget:

Object is the target of a drag and drop procedure o Flavor: The flavor describes the type of a drag and drop

description. In a drag and drop operation, you can only drop an object onto another if both have at least one common

description. o Effect: Specifies whether the drag and drop operations copies or moves the object. o Effect_In_Ctrl:

The drop effect used when you copy or move data within the same control. Process Flow of a Drag and Drop

Operation Example of Drag and Drop Programming Context Menu The context menu (right-hand mouse button or

SHIFT+F10) allows you to display a context-sensitive menu You register for the event context_menu and

context_menu_selected using the set_registered_events method. Define methods to handle both events o

Constructing a Context Menu § When you implement the handler method for the event context_menu , you must assign

the menu to the control. You may need to check the particular context in which the user requested the context menu. §

You construct the context menu using class CL_CTMENU . Almost all control wrappers pass a context menu object

reference as an event parameter of the context_menu event. If this is not the case (for example, SAP Picture), you must

create an object of the class CL_CTMENU . You can use the following methods with the context menu object:

LOAD_GUI_STATUS - Loads a context menu that you have already defined in the Menu Painter (see

below) ADD_FUNCTION - Adds a function ADD_MENU - Adds a menu that you defined in the Menu

Painter ADD_SUBMENU - Adds a menu that you defined in the Menu Painter as a submenu ADD_SEPARATOR -

Adds a separator RESET - Reset to initial value HIDE FUNCTIONS - Hides a function SHOW_FUNCTIONS

- Shows a function `DISABLE_FUNCTIONS` - Inactivates a function `ENABLE_FUNCTIONS` - Activates a function
 SAP HTML Viewer The program `SAPHTML_EVENTS_DEMO` shows how you can use the SAP HTML Viewer
 SAP Calendar Control SAP TextEdit Control ALV Grid Control SAP Container

There are five kinds of SAP Containers: SAP Custom Container

The SAP Custom Container allows you to display controls in an area defined on a normal screen using the Screen Painter.

Class: `CL_GUI_CUSTOM_CONTAINER` SAP Dialog Box Container

The SAP Dialog Box container allows you to display controls in an amodal dialog box or fullscreen.

Class: `CL_GUI_DIALOGBOX_CONTAINER` SAP Docking Container

The SAP Docking Container allows you to attach a control to any of the four edges of a screen as a resizable screen area. You can also detach it so that it becomes an independent amodal dialog box.

Class: `CL_GUI_DOCKING_CONTAINER` SAP Splitter Container

The SAP Splitter Container allows you to display more than one control in a given area by dividing it into cells.

Class: `CL_GUI_SPLITTER_CONTAINER` SAP Easy Splitter Container

The SAP Easy Splitter Container allows you to divide an area into two cells with a control in each. The cells are separated by a moveable splitter bar.

Class: `CL_GUI_EASY_SPLITTER_CONTAINER`