Tree Control and Text Editor Control

Following report displays a tree control and a text editor control in a splitter container. Double clicking on any node displays its text in editor. After editing the text double click to save it.
SET PF-STATUS 'ZSTAT1'.
IF container IS INITIAL.
   CREATE OBJECT container
      EXPORTING
         container_name = 'CONTAINER_NAME'
      EXCEPTIONS
         cntl_error = 1
         cntl_system_error = 2
         create_error = 3
         lifetime_error = 4
         lifetime_dynpro_dynpro_link = 5
         OTHERS = 6.
   IF sy-subrc <> 0.
      MESSAGE ID sy-mgid TYPE sy-msgty NUMBER sy-msgno
         WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
   ENDIF.
   CREATE OBJECT splitter
      EXPORTING
         parent = container
         orientation = 1
         name = 'Mohit'
      EXCEPTIONS
         cntl_error = 1
         cntl_system_error = 2
         OTHERS = 3.
   IF sy-subrc <> 0.
      MESSAGE ID sy-mgid TYPE sy-msgty NUMBER sy-msgno
         WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
   ENDIF.
   left = splitter->top_left_container.
   right = splitter->bottom_right_container.
   CREATE OBJECT editor
      EXPORTING
         parent = right
         name = 'MohitEditor'
      EXCEPTIONS
         error_cntl_create = 1
         error_cntl_init = 2
         error_cntl_link = 3
         error_dp_create = 4
         gui_type_not_supported = 5
         OTHERS = 6.
   IF sy-subrc <> 0.
      MESSAGE ID sy-mgid TYPE sy-msgty NUMBER sy-msgno
         WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
   ENDIF.
   CREATE OBJECT tree
      EXPORTING
         parent = left
         node_selection_mode = tree->node_sel_mode_single
         name = 'MohitTree'
      EXCEPTIONS
         lifetime_error = 1
cntl_system_error = 2
create_error = 3
failed = 4
illegal_node_selection_mode = 5
OTHERS = 6.

IF sy-subrc <> 0.
  MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
  WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ENDIF.

PERFORM fill_tree.

CALL METHOD tree->add_nodes
  EXPORTING
    table_structure_name = 'NODE_STR'
    node_table = node_itab

EXCEPTIONS
  error_in_node_table = 1
  failed = 2
  dp_error = 3
  table_structure_name_not_found = 4
  OTHERS = 5.

IF sy-subrc <> 0.
  MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
  WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ENDIF.

CREATE OBJECT handler1.
PERFORM register_events.
PERFORM register_events_textedit.
ENDIF.

ENDMODULE. " start OUTPUT

*-----------------------------------------------*
* Module USER_COMMAND_9001 INPUT                 *
*-----------------------------------------------*
* text                                         *
*-----------------------------------------------*

MODULE user_command_9001 INPUT.
  CALL METHOD cl_gui_cfw=>dispatch.
ENDMODULE. " USER_COMMAND_9001 INPUT

*-----------------------------------------------*
* Module USER_COMMAND_9001 INPUT                 *
*-----------------------------------------------*
* text                                         *
*-----------------------------------------------*

FORM fill_tree .
  data: node LIKE node_str.
  CLEAR node.
  node-node_key = 'head_top'.
  node-isfolder = 'X'.
  node-text = 'Head1'.
  APPEND node TO node_itab.
  CLEAR node.
  node-node_key = 'Child1'.
  node-relatkey = 'head_top'.
  node-relatship = cl_gui_simple_tree=>relat_last_child.
  node-text = 'First '.
  APPEND node TO node_itab.
CLEAR node.
ode-node_key = 'Child2'.
ode-relatkey = 'head_top'.
ode-relatship = cl_gui_simple_tree=>relat_last_child.
ode-text = 'Second '.
APPEND node TO node_itab.
CLEAR node.
ode-node_key = 'head_item'.
ode-isfolder = 'X'.
ode-text = 'Third'.
APPEND node TO node_itab.
CLEAR node.
ode-node_key = 'Child3'.
ode-relatkey = 'head_item'.
ode-relatship = cl_gui_simple_tree=>relat_next_sibling.
ode-text = ' INDIA '.
APPEND node TO node_itab.
CLEAR node.
ode-node_key = 'Child4'.
ode-relatkey = 'head_item'.
ode-relatship = cl_gui_simple_tree=>relat_last_child.
ode-text = 'SAP '.
APPEND node TO node_itab.
ENDFORM. " fill_tree
*\---------------------------------------------------------------------*
*\ Form node_double_click
*\---------------------------------------------------------------------*
FORM node_double_click USING p_node_key TYPE tv_nodekey.
    CLEAR : v_node_key.
    DATA node LIKE node_str.
    DATA textline(256).
    DATA text_table LIKE STANDARD TABLE OF textline.
    v_node_key = p_node_key.
    READ TABLE node_itab WITH KEY node_key = p_node_key INTO node.
    IF sy-subrc = 0.
        MOVE node-text TO textline.
        APPEND textline TO text_table.
        CALL METHOD editor->set_text_as_stream
        EXPORTING
            text = text_table
        EXCEPTIONS
            error_dp = 1
            error_dp_create = 2
            others = 3.
        IF sy-subrc <> 0.
            MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
            WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
        ENDIF.
    ENDIF.
ENDFORM. " node_double_click
*\---------------------------------------------------------------------*
*\ Module exit INPUT
MODULE exit INPUT.
    CASE sy-ucomm.
        WHEN 'EXIT'.
            LEAVE PROGRAM.
    ENDCASE.
ENDMODULE. " exit INPUT

FORM register_events.
    DATA: lt_events TYPE cntl_simple_events,
        l_event TYPE cntl_simple_event.
    CALL METHOD tree->get_registered_events
        IMPORTING
            events = lt_events
        EXCEPTIONS
            cntl_error = 1
            OTHERS = 2.
    IF sy-subrc <> 0.
        MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
            WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
    ENDIF.
    l_event-eventid = cl_gui_simple_tree=>eventid_node_double_click.
    APPEND l_event TO lt_events.
    CALL METHOD tree->set_registered_events
        EXPORTING
            events = lt_events
        EXCEPTIONS
            cntl_error = 1
            cntl_system_error = 2
            illegal_event_combination = 3
            OTHERS = 4.
    IF sy-subrc <> 0.
        MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
            WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
    ENDIF.
    SET HANDLER handler1->handle_node_double_click FOR tree.
ENDFORM. " register_events

FORM register_events_textedit.
    DATA: lt_events TYPE cntl_simple_events,
        l_event TYPE cntl_simple_event.
    CALL METHOD editor->get_registered_events
        IMPORTING
            events = lt_events
        EXCEPTIONS
            cntl_error = 1
           OTHERS = 2.
IF sy-subrc <> 0.
    MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
    WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ENDIF.

l_event-eventid = cl_gui_textedit->event_double_click.
APPEND l_event TO lt_events.
CALL METHOD editor->set_registered_events
  EXPORTING
    events = lt_events
EXCEPTIONS
    cntl_error = 1
    cntl_system_error = 2
    illegal_event_combination = 3
    OTHERS = 4.
IF sy-subrc <> 0.
    MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
    WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ENDIF.

SET HANDLER handler1->handle_double_click FOR editor.
ENDFORM. " register_events_textedit

*---------------------------------------------------------------------*
* Form handle_double_click                                            *
*---------------------------------------------------------------------*

FORM handle_double_click .
  DATA node LIKE node_str.
  DATA textline1(256).
  DATA text_table1 LIKE STANDARD TABLE OF textline1.
  CALL METHOD editor->get_text_as_stream
    EXPORTING
      only_when_modified = 1
    IMPORTING
      text = text_table1
EXCEPTIONS
    error_dp = 1
    error_cntl_call_method = 2
    OTHERS = 3.
IF sy-subrc <> 0.
    MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
    WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ENDIF.

IF text_table1 IS NOT INITIAL.
  READ TABLE text_table1 INTO textline1 INDEX 1.
  READ TABLE node_itab INTO node WITH KEY node_key = v_node_key.
  IF sy-subrc = 0.
    node-text = textline1.
    MODIFY node_itab FROM node INDEX sy-tabix.
  ENDIF.
END.
CALL METHOD editor->set_text_as_stream
  EXPORTING
    text = text_table1
EXCEPTIONS
    error_dp = 1
    error_dp_create = 2
OTHERS = 3.
IF sy-subrc <> 0.
  MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
  WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ENDIF.
CALL METHOD tree->delete_all_nodes
  EXCEPTIONS
    failed = 1
    cntl_system_error = 2
    OTHERS = 3.
IF sy-subrc <> 0.
  MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
  WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ENDIF.
CALL METHOD tree->add_nodes
  EXPORTING
    table_structure_name = 'NODE_STR'
    node_table = node_itab
  EXCEPTIONS
    error_in_node_table = 1
    failed = 2
    dp_error = 3
    table_structure_name_not_found = 4
    OTHERS = 5.
IF sy-subrc <> 0.
  MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
  WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ENDIF.
MESSAGE s000.
ENDIF.
ENDDFORM. " handle_double_click