

第九章 Tree Control 的使用

Tree Control 是最常用的 Windows 控件之一，在其他语言中，称为“Tree View”等，ABAP 的 Tree Control 能完成相似的功能，本章主要介绍以下实例：

- (1) 实现一个简单的 Tree Control；
- (2) 建立有数据表接口的 Tree Control 对象；
- (3) 通过右键菜单完成 Tree Control 节点的增删修改等功能；
- (4) Tree Control 对象与 ALV Grid 对象关联，实现在双击 Tree 节点时，变更 ALV Grid 的显示。

各例子简捷独立，先简单后复杂。

9.1 简单的 Tree Control 实例

如图 9-1 所示，本例将实现以下功能：

- (1) 建立一个以国家、城市、城区三层的 Tree Control 对象（固定数据）；
- (2) 当双击某个节点时，在文本框显示点的节点的 KEY 值和节点文本值。

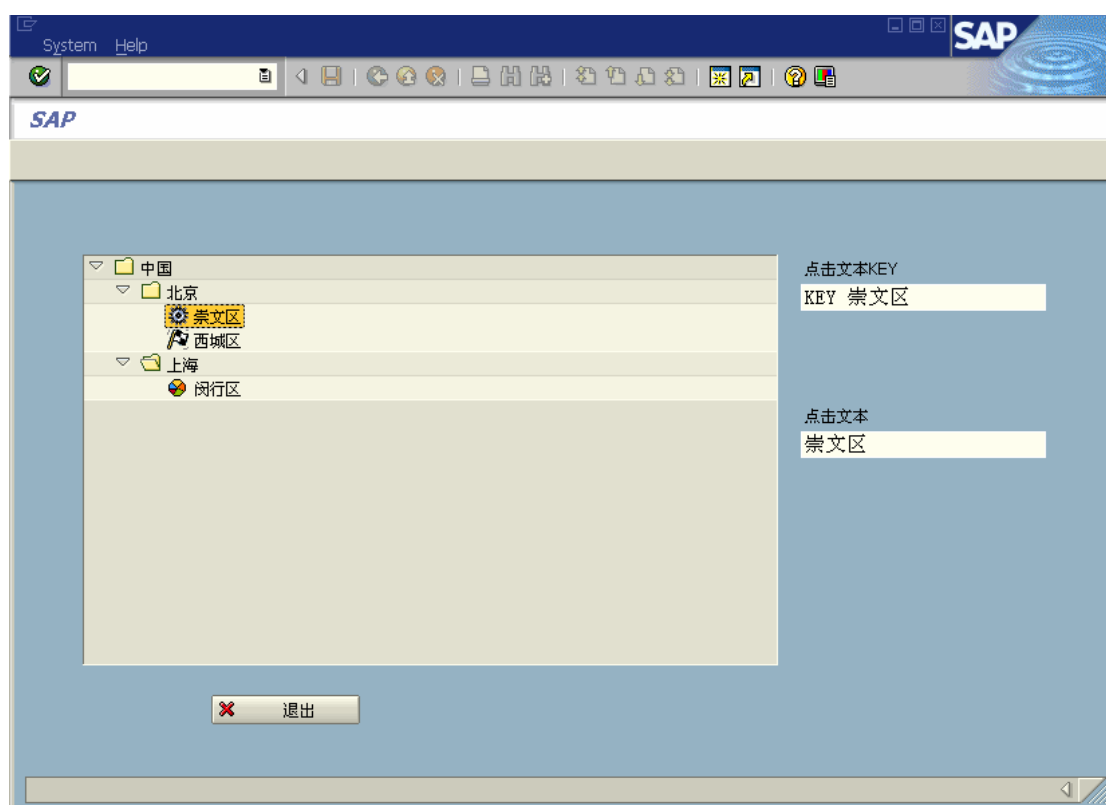


图 9-1

实现步骤如下（前两步为基本功能，不再详细描述）：

- (1) 建立一个 ABAP 程序；
- (2) 使用屏幕制作器，建立 Screen。
 - 添加“定制控制”对象；

- 添加两个文本框，存放节点 KEY 值和节点文本值。

(3) 建立程序。

- 处理窗口调用，退出等基本功能；
- 创建窗口定制对象，Tree Control 对象，测试运行时能看到 TREE 区域
- 实现 Tree 节点的创建，测试运行，能看到 Tree 的所有节点；
- 定义对象，建立双击方法，并将双击方法分配 Tree 对象，实现双击事件。

建立好的屏幕和对象界面如图 9-2 所示。

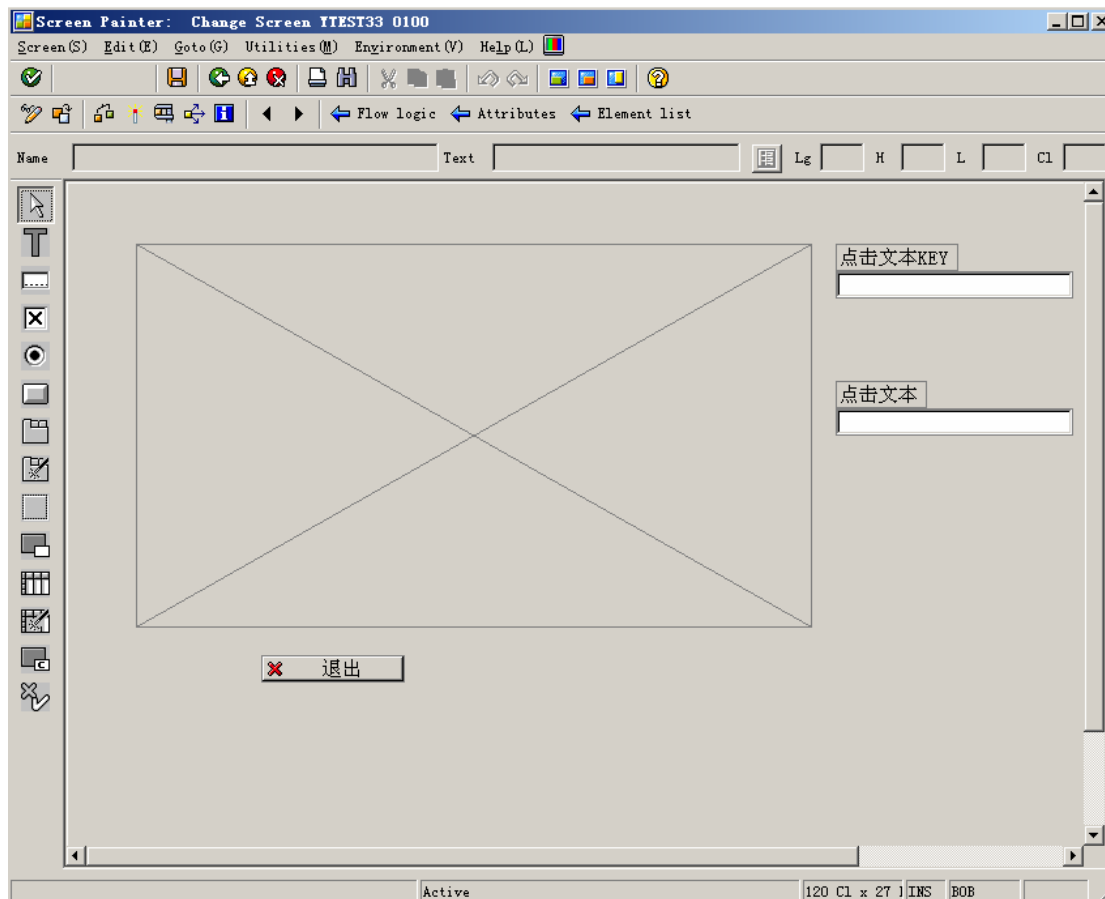


图 9-2

元素清单如图 9-3 所示。

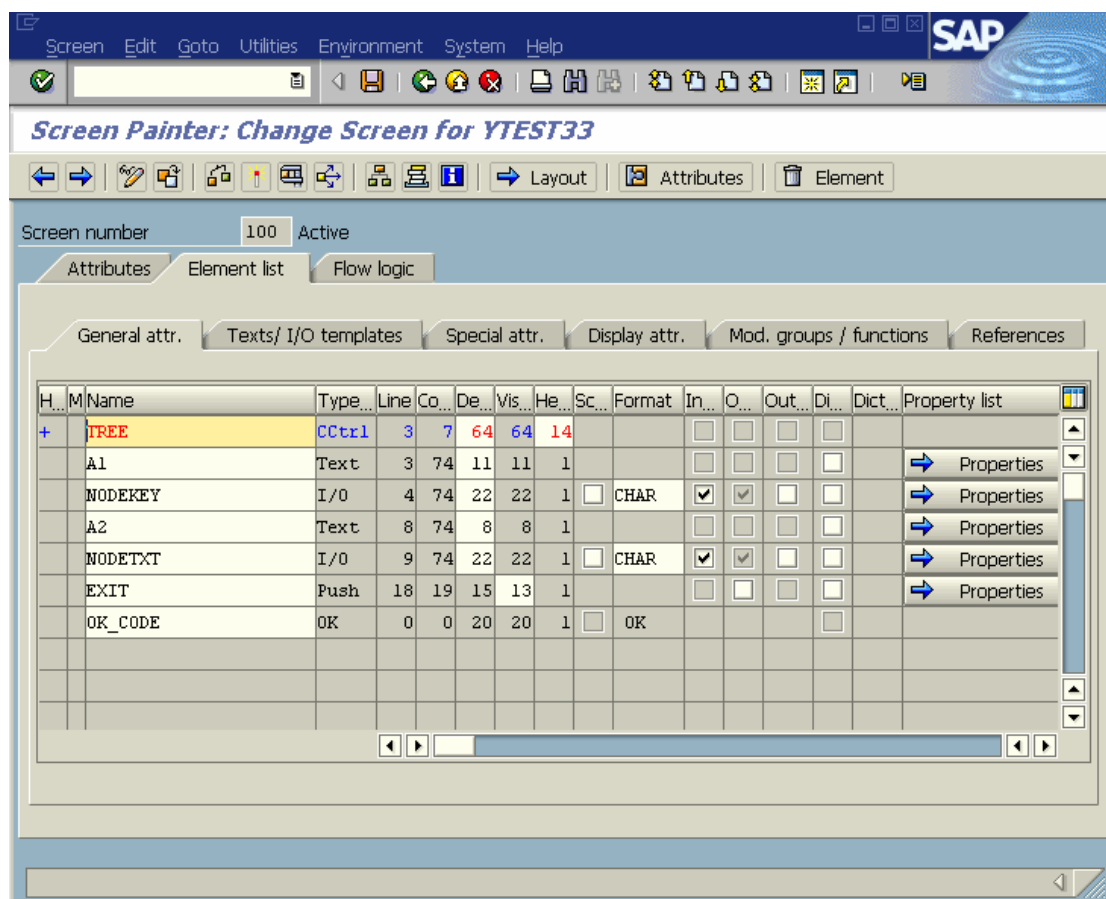


图 9-3

逻辑程序:

*PBO显示屏幕前的处理

PROCESS BEFORE OUTPUT.

MODULE STATUS_0100.

*PAI用户输入后的处理

PROCESS AFTER INPUT.

MODULE USER_COMMAND_0100.

主程序:

REPORT YTEST33.

*功能码返回值

DATA: OK_CODE TYPE SY-UCOMM,

SAVE_OK TYPE SY-UCOMM.

*定制对象

DATA: WA_CONTAINER TYPE SCRFNAME VALUE 'TREE',

*客户对象

WA_CUSTOM_CONTAINER TYPE REF TO CL_GUI_CUSTOM_CONTAINER,

WA_TREE TYPE REF TO CL_GUI_SIMPLE_TREE.

*NODE 节点表

DATA: NODE_TABLE LIKE TABLE OF MTREESNODE,

*节点

NODE1 TYPE MTREESNODE.

*定义内表, 变量需要传递, 不加HEADER LINE

DATA WA_SPFLI TYPE TABLE OF SPFLI.

*当双击时取得节点KEY值与节点文本值变量

DATA: NODEKEY(200) VALUE 'NODE', NODETEXT(200) VALUE 'NONE'.

*定义对象, 定义对象的双击方法, 用于将该对象双击方法分配到树双击事件

CLASS LCL_APPLICATION DEFINITION DEFERRED.

* CLASS lcl_application DEFINITION

* 对象定义

CLASS LCL_APPLICATION DEFINITION.

PUBLIC SECTION.

*自定义双击方法, 参数为节点关键字

METHODS HANDLE_NODE_DOUBLE_CLICK

FOR EVENT NODE_DOUBLE_CLICK

OF CL_GUI_SIMPLE_TREE

IMPORTING NODE_KEY.

ENDCLASS. "

* CLASS lcl_application IMPLEMENTATION

* 对象方法实现

CLASS LCL_APPLICATION IMPLEMENTATION.

METHOD HANDLE_NODE_DOUBLE_CLICK.

* 当双击时弹出信息窗口, 调试用

* MESSAGE I208(00) WITH 'double'.

*取得KEY值, 在窗口显示

NODEKEY = NODE_KEY.

*从节点内表中按关键字读取单个节点

READ TABLE NODE_TABLE WITH KEY NODE_KEY = NODE_KEY INTO NODE1.

*将节点文本在窗口中显示

NODETEXT = NODE1-TEXT.

ENDMETHOD.

"HANDLE_NODE_DOUBLE_CLICK

ENDCLASS. "lcl_application IMPLEMENTATION

*自定义定义对象实例

DATA: G_APPLICATION TYPE REF TO LCL_APPLICATION.

START-OF-SELECTION.

CREATE OBJECT G_APPLICATION.

* 直接调用窗口, 在窗口PBO分配双击对象

CALL SCREEN 100.

&-----

*& Module STATUS_0100 OUTPUT

&-----

* text

MODULE STATUS_0100 OUTPUT.

* SET PF-STATUS 'STATUS1'.

IF WA_CUSTOM_CONTAINER IS INITIAL.

PERFORM CREATE_TREE.

ENDIF.

ENDMODULE. " STATUS_0100 OUTPUT

&-----

*& Module USER_COMMAND_0100 INPUT

&-----

* 用户交互

MODULE USER_COMMAND_0100 INPUT.

SAVE_OK = OK_CODE.

CLEAR OK_CODE.

CASE SAVE_OK.

WHEN 'EXIT'.

LEAVE PROGRAM.

ENDCASE.

ENDMODULE. " USER_COMMAND_0100 INPUT

&-----

*& Form CREATE_TREE

&-----

* 创建树及子节点

* --> p1 text

* <-- p2 text

FORM CREATE_TREE .

*事件内表及单个事件对象

DATA: EVENTS TYPE CNTL_SIMPLE_EVENTS,

EVENT1 TYPE CNTL_SIMPLE_EVENT.

*建立定制控制对象

```

CREATE OBJECT WA_CUSTOM_CONTAINER
EXPORTING
CONTAINER_NAME = WA_CONTAINER.
*建立树对象
CREATE OBJECT WA_TREE
EXPORTING
*   LIFETIME           =
PARENT           = WA_CUSTOM_CONTAINER
*   SHELLSYL           =
NODE_SELECTION_MODE = CL_GUI_SIMPLE_TREE=>NODE_SEL_MODE_SINGLE
*   HIDE_SELECTION     =
*   NAME               =
*   EXCEPTIONS
*   LIFETIME_ERROR     = 1
*   CNTL_SYSTEM_ERROR = 2
*   CREATE_ERROR       = 3
*   FAILED             = 4
*   ILLEGAL_NOCE_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.
*开始建立以国家-> 城市->城区 三层 的树
CLEAR NODE1.
*节点关键字
NODE1-NODE_KEY = '国家'.
*第一层
CLEAR NODE1-RELATKEY.
CLEAR NODE1-RELATSHIP.
*是文件夹
NODE1-HIDDEN = ''.
NODE1-DISABLED = ''.
NODE1-ISFOLDER = 'X'.
*不带ICON
CLEAR NODE1-N_IMAGE.
CLEAR NODE1-EXP_IMAGE.
CLEAR NODE1-EXPANDER.

NODE1-TEXT = '中国'.
APPEND NODE1 TO NODE_TABLE.
*二层节点
CLEAR NODE1.

```

```
NODE1-NODE_KEY = '城市北京'.
```

***上层关系**

```
NODE1-RELATKEY = '国家'.
NODE1-RELATSHIP = CL_GUI_SIMPLE_TREE=>RELAT_LAST_CHILD.
NODE1-HIDDEN = ''.
NODE1-DISABLED = ''.
NODE1-ISFOLDER = 'X'.
CLEAR NODE1-N_IMAGE.
CLEAR NODE1-EXP_IMAGE.
CLEAR NODE1-EXPANDER.
NODE1-TEXT = '北京'.
APPEND NODE1 TO NODE_TABLE.
```

```
CLEAR NODE1.
```

```
NODE1-NODE_KEY = '城市上海'.
NODE1-RELATKEY = '国家'.
NODE1-RELATSHIP = CL_GUI_SIMPLE_TREE=>RELAT_LAST_CHILD.
NODE1-HIDDEN = ''.
NODE1-DISABLED = ''.
NODE1-ISFOLDER = 'X'.
CLEAR NODE1-N_IMAGE.
CLEAR NODE1-EXP_IMAGE.
CLEAR NODE1-EXPANDER.
NODE1-TEXT = '上海'.
APPEND NODE1 TO NODE_TABLE.
```

```
CLEAR NODE1.
NODE1-NODE_KEY = 'KEY 崇文区'.
NODE1-RELATKEY = '城市北京'.
NODE1-RELATSHIP = CL_GUI_SIMPLE_TREE=>RELAT_LAST_CHILD.
NODE1-HIDDEN = ''.
NODE1-DISABLED = ''.
NODE1-ISFOLDER = ''.
```

***图标, 可运行SHOWICON查阅**

```
NODE1-N_IMAGE = '@9Y@'.
NODE1-EXP_IMAGE = '@9Y@'.
CLEAR NODE1-EXPANDER.
NODE1-TEXT = '崇文区'.
APPEND NODE1 TO NODE_TABLE.
```

```
CLEAR NODE1.
NODE1-NODE_KEY = 'KEY 西城区'.
```

```

NODE1-RELATKEY = '城市北京'.
NODE1-RELATSHIP = CL_GUI_SIMPLE_TREE=>RELAT_LAST_CHILD.
NODE1-HIDDEN = ''.
NODE1-DISABLED = ''.
NODE1-ISFOLDER = ''.
NODE1-N_IMAGE = '@DF@'.
NODE1-EXP_IMAGE = '@DF@'.
CLEAR NODE1-EXPANDER.
NODE1-TEXT = '西城区'.
APPEND NODE1 TO NODE_TABLE.

CLEAR NODE1.
NODE1-NODE_KEY = 'KEY 闵行区'.
NODE1-RELATKEY = '城市上海'.
NODE1-RELATSHIP = CL_GUI_SIMPLE_TREE=>RELAT_LAST_CHILD.
NODE1-HIDDEN = ''.
NODE1-DISABLED = ''.
NODE1-ISFOLDER = ''.
NODE1-N_IMAGE = '@NL@'.
NODE1-EXP_IMAGE = '@NL@'.
CLEAR NODE1-EXPANDER.
NODE1-TEXT = '闵行区'.
APPEND NODE1 TO NODE_TABLE.
*按照节点内容添加节点
CALL METHOD WA_TREE->ADD_NODES
EXPORTING
    TABLE_STRUCTURE_NAME = 'MTREESNODE'
    NODE_TABLE             = NODE_TABLE
* EXCEPTIONS
*   ERROR_IN_NODE_TABLE   = 1
*   FAILED                = 2
*   DP_ERROF              = 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.
*定义双击事件
EVENT1-EVENTID = CL_GUI_SIMPLE_TREE=>EVENTID_NODE_DOUBLE_CLICK.
EVENT1-APPL_EVENT = 'X'.
APPEND EVENT1 TO EVENTS.
*添加事件内表

```



```

CALL METHOD WA_TREE->SET_REGISTERED_EVENTS
EXPORTING
  EVENTS = 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.
*将已定义的双击事件分配至树对象
SET HANDLER G_APPLICATION->HANDLE_NODE_DOUBLE_CLICK FOR WA_TREE.
ENDFORM.                " CREATE_TREE
    
```

9.2 建立有数据表接口的 Tree Control 对象

先从数据表读取结构，并按照数据关系建立 Tree Control，如图 9-4 所示。

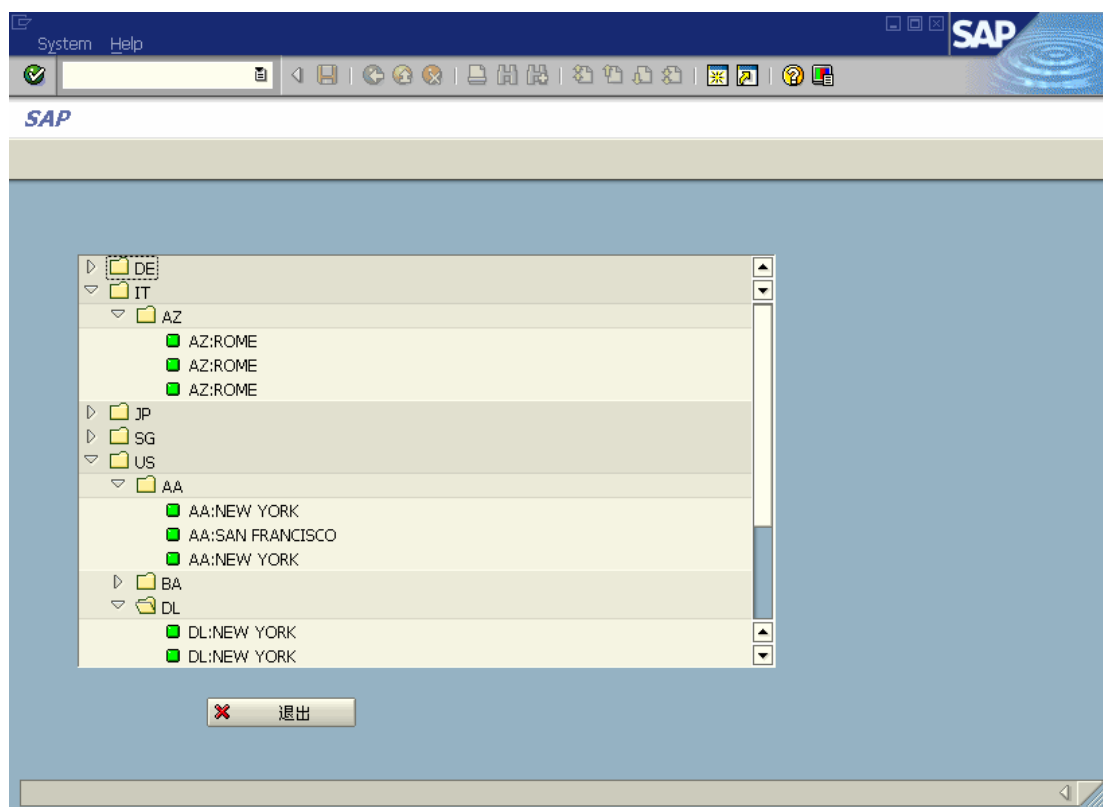


图 9-4

以航班表为例，以“起飞国家—>航班承运人—>航班连接”三层组成。

主程序：

REPORT YTEST34 .

***功能码返回值**

DATA: OK_CODE TYPE SY-UCOMM,
 SAVE_OK TYPE SY-UCOMM.

***定制对象**

DATA: WA_CONTAINER TYPE SCRFNAME VALUE 'TREE',

***客户对象**

 WA_CUSTOM_CONTAINER TYPE REF TO CL_GUI_CUSTOM_CONTAINER,
 WA_TREE TYPE REF TO CL_GUI_SIMPLE_TREE.

***NODE 节点表**

DATA: NODE_TABLE LIKE TABLE OF MTREESNODE,

***节点**

 NODE1 TYPE MTREESNODE.

DATA WA_SPFLI TYPE TABLE OF SPFLI WITH HEADER LINE.

START-OF-SELECTION.

 SELECT * INTO TABLE WA_SPFLI FROM SPFLI
 ORDER BY COUNTRYFR CARRID CONNID ASCENDING.

 SET SCREEN 100.

&-----

*& Module STATUS_0100 OUTPUT

&-----

* text

MODULE STATUS_0100 OUTPUT.

* SET PF-STATUS 'STATUS1'.

* CLEAR WA_CUSTOM_CONTAINER.

 IF WA_CUSTOM_CONTAINER IS INITIAL.

 PERFORM CREATE_TREE.

 ENDIF.

ENDMODULE. " STATUS_0100 OUTPUT

&-----

*& Module USER_COMMAND_0100 INPUT

&-----

* 用户交互

MODULE USER_COMMAND_0100 INPUT.

 SAVE_OK = OK_CODE.

 CLEAR OK_CODE.

 CASE SAVE_OK.

 WHEN 'EXIT'.

```

*      LEAVE PROGRAM.
      LEAVE TO SCREEN 0.
    ENDCASE.
ENDMODULE.                                " USER_COMMAND_0100  INPUT
*&-----*
*&      Form  CREATE_TREE
*&-----*
*      创建树及子节点
*-----*
*  -->  p1          text
*  <--  p2          text
*-----*
FORM CREATE_TREE .
*建立定制控制对象
  CREATE OBJECT WA_CUSTOM_CONTAINER
    EXPORTING
      CONTAINER_NAME = WA_CONTAINER.
*建立树对象
  CREATE OBJECT WA_TREE
    EXPORTING
*    LIFETIME          =
*    PARENT            = WA_CUSTOM_CONTAINER
*    SHELLSYLE        =
*    NODE_SELECTION_MODE = CL_GUI_SIMPLE_TREE=>NODE_SEL_MODE_SINGLE
*    HIDE_SELECTION    =
*    NAME              =
*    EXCEPTIONS
*    LIFETIME_ERROR    = 1
*    CNTL_SYSTEM_ERROR = 2
*    CREATE_ERROR      = 3
*    FAILED            = 4
*    ILLEGAL_NOCE_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.
  DATA: COUNTRYFR LIKE SPFLI-COUNTRYFR,
        CARRID LIKE SPFLI-CARRID.

  DATA: STR1(255).

  LOOP AT WA_SPFLI.

```

```

    IF COUNTRYFR NE WA_SPFLI-COUNTRYFR.
        CLEAR NODE1.
        NODE1-NODE_KEY = WA_SPFLI-COUNTRYFR.
*第一层
        CLEAR NODE1-RELATKEY.
        CLEAR NODE1-RELATSHIP.
*是文件夹
        NODE1-HIDDEN = ''.
        NODE1-DISABLED = ''.
        NODE1-ISFOLDER = 'X'.
*不带ICON
        CLEAR NODE1-N_IMAGE.
        CLEAR NODE1-EXP_IMAGE.
        CLEAR NODE1-EXPANDER.

        NODE1-TEXT = WA_SPFLI-COUNTRYFR.
        APPEND NODE1 TO NODE_TABLE.
    ENDIF.

    IF ( COUNTRYFR NE WA_SPFLI-COUNTRYFR ) OR
        ( CARRID <> WA_SPFLI-CARRID ).
        CLEAR NODE1.
        CONCATENATE WA_SPFLI-COUNTRYFR WA_SPFLI-CARRID
            INTO NODE1-NODE_KEY.
*上层关系
        NODE1-RELATKEY = WA_SPFLI-COUNTRYFR.
        NODE1-RELATSHIP = CL_GUI_SIMPLE_TREE=>RELAT_LAST_CHILD.
        NODE1-HIDDEN = ''.
        NODE1-DISABLED = ''.
        NODE1-ISFOLDER = 'X'.
        CLEAR NODE1-N_IMAGE.
        CLEAR NODE1-EXP_IMAGE.
        CLEAR NODE1-EXPANDER.
*节点显示CARRID
        NODE1-TEXT = WA_SPFLI-CARRID.
        APPEND NODE1 TO NODE_TABLE.
    ENDIF.

    COUNTRYFR = WA_SPFLI-COUNTRYFR.
    CARRID = WA_SPFLI-CARRID.
    CLEAR NODE1.
    CONCATENATE WA_SPFLI-COUNTRYFR WA_SPFLI-CARRID
        WA_SPFLI-CONNID INTO NODE1-NODE_KEY.

```

```

CONCATENATE WA_SPFLI-COUNTRYFR WA_SPFLI-CARRID
      INTO STR1.
NODE1-RELATKEY = STR1.
NODE1-RELATSHIP = CL_GUI_SIMPLE_TREE=>RELAT_LAST_CHILD.
NODE1-HIDDEN = ''.
NODE1-DISABLED = ''.
NODE1-ISFOLDER = 'X'.
NODE1-N_IMAGE = '@5B@'.
NODE1-EXP_IMAGE = '@5B@'.
CLEAR NODE1-EXPANDER.
CONCATENATE WA_SPFLI-CARRID ':' WA_SPFLI-CITYFROM
      INTO STR1.
NODE1-TEXT = STR1.
APPEND NODE1 TO NODE_TABLE.
ENDLOOP.

```

*按照节点内容添加节点

```

CALL METHOD WA_TREE->ADD_NODES
      EXPORTING
        TABLE_STRUCTURE_NAME      = 'MTREESNODE'
        NODE_TABLE                  = NODE_TABLE
*   EXCEPTIONS
*     ERROR_IN_NODE_TABLE          = 1
*     FAILED                      = 2
*     DP_ERROF                    = 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.
ENDFORM.                " CREATE_TREE

```

9.3 通过右键菜单完成 Tree Control 节点的增删改功能

Tree 节点的增删改是很重要的功能, 包括: 增加本级节点, 增加子节点, 修改节点, 删除节点。完成后效果如下:

选择根节点, 单击右键, 弹出右键菜单, 如选择“北京”节点, 选择“增加本级节点”菜单, 如图 9-5 所示。

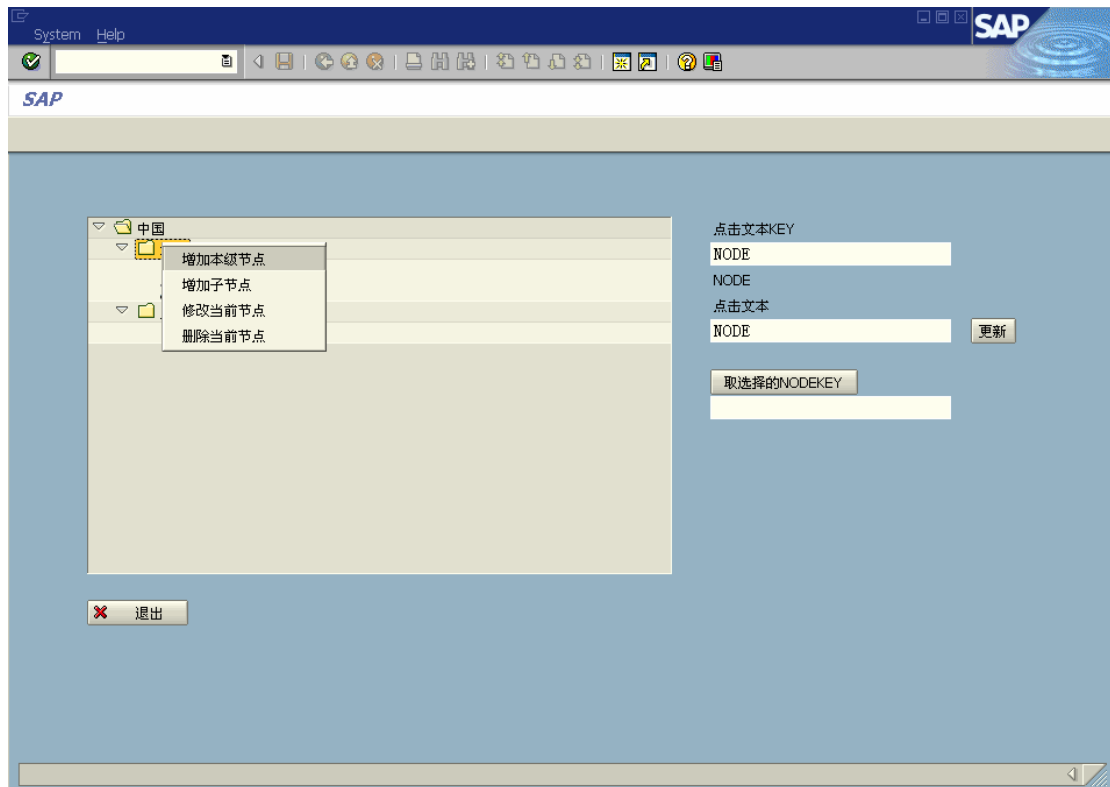


图 9-5

在单击文本输入“广州”后，单击“更新”按钮，如图 9-6 所示。

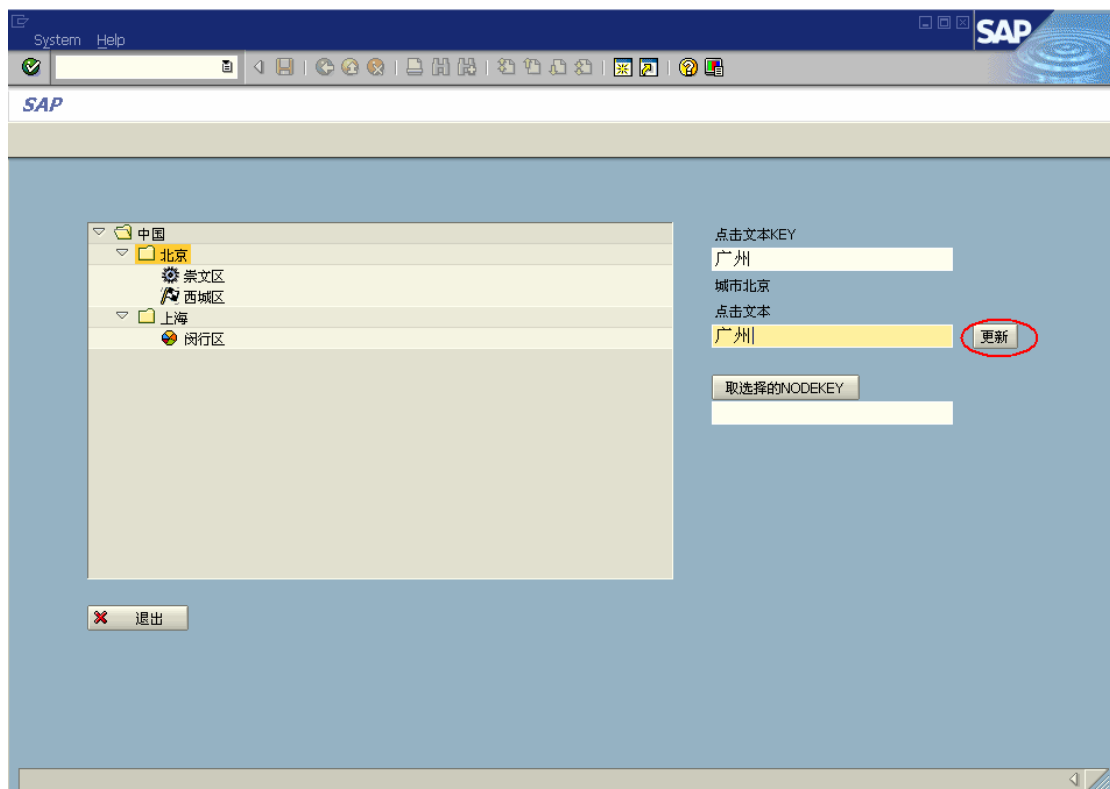


图 9-6

Tree 对象增加了“广州”节点，选择该节点，选右键菜单中“增加子节点”菜单，如图 9-7 所示。

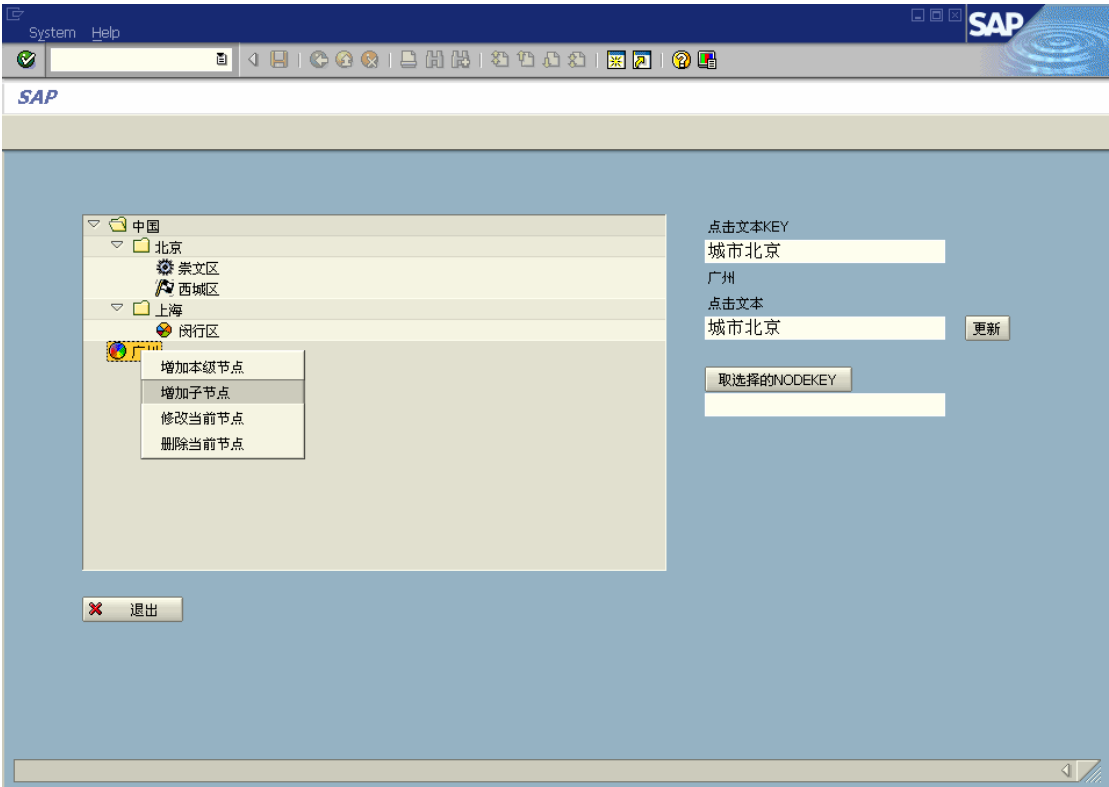


图 9-7

输入“天河区”，单击“更新”按钮，如图 9-8 所示。

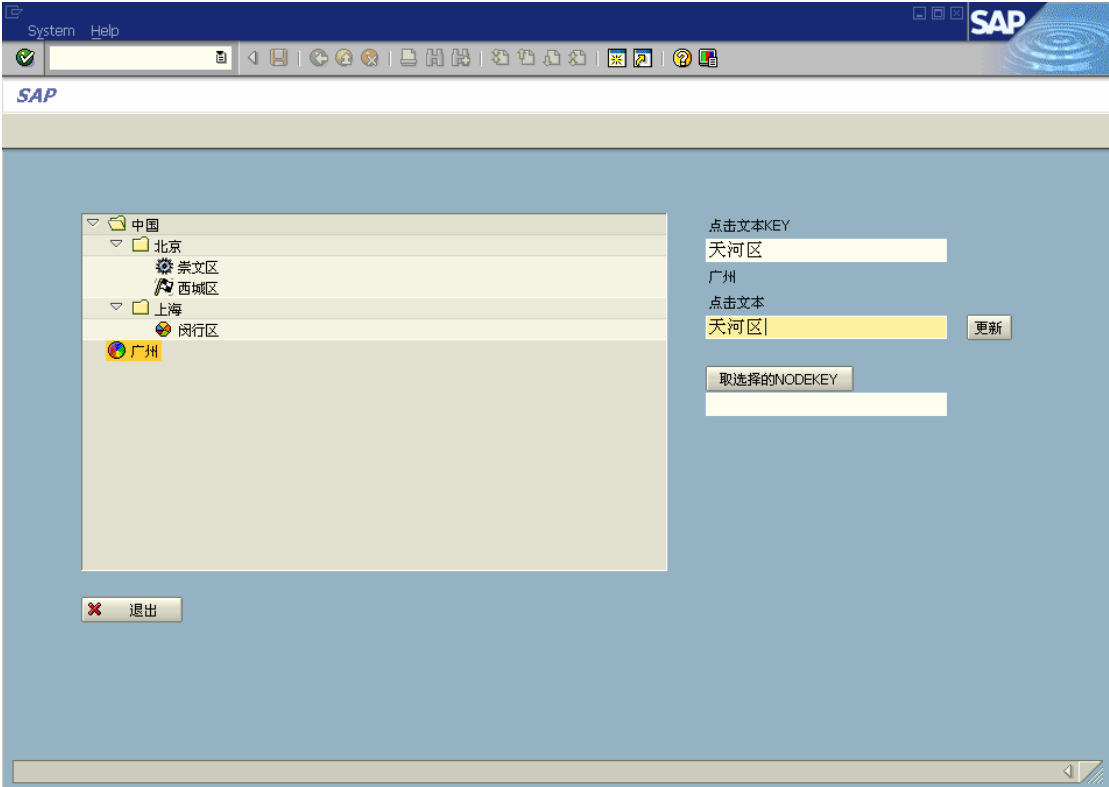


图 9-8

增加节点后，输出结果如图 9-9 所示。

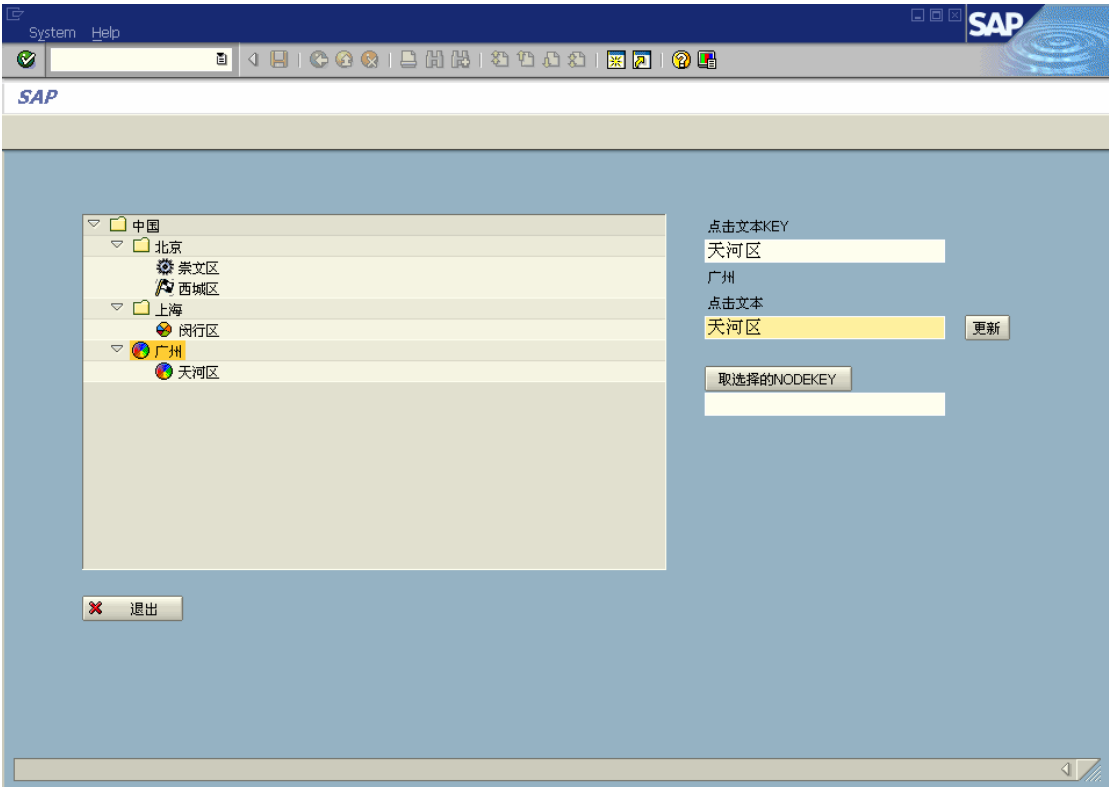


图 9-9

除上述功能外，还有修改及删除节点等功能。
设计 Screen 界面如图 9-10 所示。

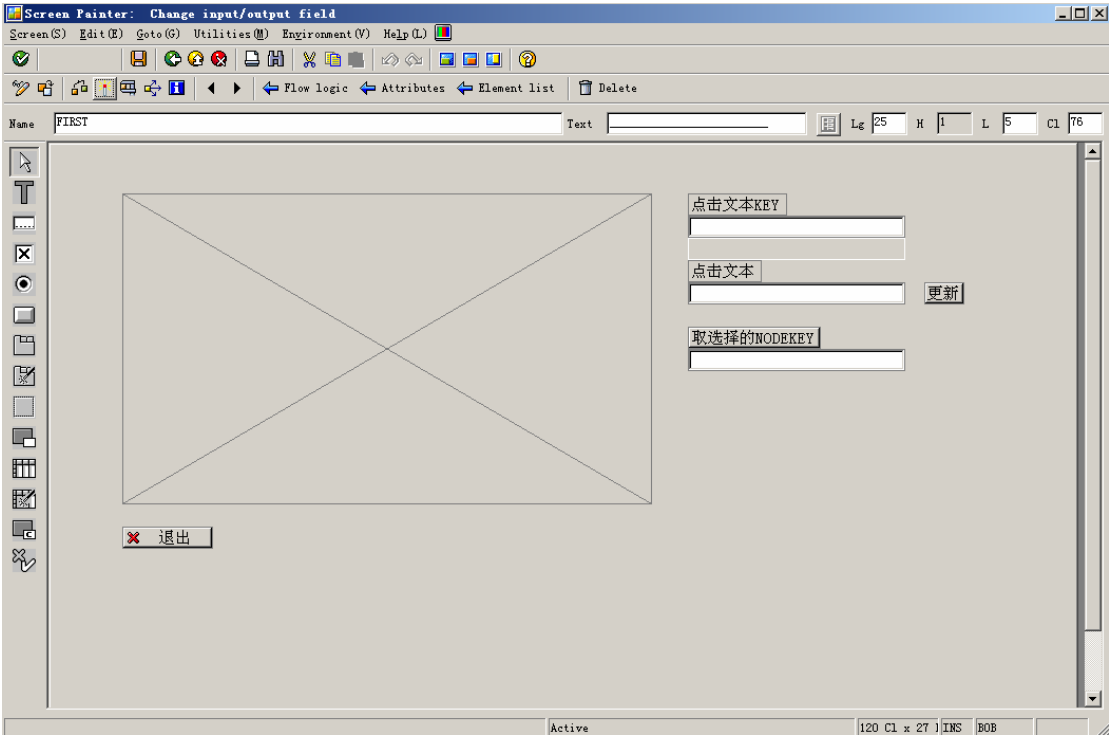


图 9-10

主程序:
REPORT YTEST35.
DATA: OK_CODE TYPE SY-UCOMM,


```

SAVE_OK TYPE SY-UCOMM.

DATA: WA_CONTAINER TYPE SCRFNAME VALUE 'TREE1',
      WA_CUSTOM_CONTAINER TYPE REF TO CL_GUI_CUSTOM_CONTAINER,
      WA_TREE TYPE REF TO CL_GUI_SIMPLE_TREE.

DATA: NODE_TABLE LIKE TABLE OF MTREESNODE,
      ANODE_TABLE LIKE TABLE OF MTREESNODE,
      NODE1 TYPE MTREESNODE,
      NODE2 TYPE MTREESNODE.

DATA WA_SPFLI TYPE TABLE OF SPFLI.

DATA: NODEKEY(200) VALUE 'NODE',
      NODETEXT(200) VALUE 'NODE',
      NODEKEYOLD(200) VALUE 'NODE',
      GETKEY(200).
DATA MENUSTR(10) VALUE ''.
CLASS LCL_APPLICATION DEFINITION DEFERRED.

*-----*
*      CLASS lcl_application DEFINITION
*-----*
*
*-----*
CLASS LCL_APPLICATION DEFINITION.
  PUBLIC SECTION.
    METHODS: HANDLE_NODE_DOUBLE_CLICK
              FOR EVENT NODE_DOUBLE_CLICK
              OF CL_GUI_SIMPLE_TREE
              IMPORTING NODE_KEY.
    METHODS:
      HANDLE_NODE_CONTEXT_MENU_REQ
      FOR EVENT NODE_CONTEXT_MENU_REQUEST
      OF CL_GUI_SIMPLE_TREE
      IMPORTING NODE_KEY MENU,
      HANDLE_NODE_CONTEXT_MENU_SEL
      FOR EVENT NODE_CONTEXT_MENU_SELECT
      OF CL_GUI_SIMPLE_TREE
      IMPORTING NODE_KEY FCODE.

ENDCLASS.
" LCL_APPLICATION DEFINITION

*-----*

```

```

*          CLASS lcl_application IMPLEMENTATION
*-----*
*
*-----*
CLASS LCL_APPLICATION IMPLEMENTATION.
  METHOD HANDLE_NODE_DOUBLE_CLICK.
    NODEKEY = NODE_KEY.
    READ TABLE NODE_TABLE WITH KEY NODE_KEY = NODE_KEY INTO NODE1.
    NODETEXT = NODE1-TEXT.
  ENDMETHOD.                                "HANDLE_NODE_DOUBLE_CLICK

  METHOD HANDLE_NODE_CONTEXT_MENU_REQ.
    CALL METHOD MENU->ADD_FUNCTION
      EXPORTING
        TEXT  = '增加本级节点'
        FCODE = 'CODE1'.

    CALL METHOD MENU->ADD_FUNCTION
      EXPORTING
        TEXT  = '增加子节点'
        FCODE = 'CODE2'.

    CALL METHOD MENU->ADD_FUNCTION
      EXPORTING
        TEXT  = '修改当前节点'
        FCODE = 'CODE3'.
    CALL METHOD MENU->ADD_FUNCTION
      EXPORTING
        TEXT  = '删除当前节点'
        FCODE = 'CODE4'.
  ENDMETHOD.                                "handle_node_context_menu_req

  METHOD HANDLE_NODE_CONTEXT_MENU_SEL.
    DATA STR1 TYPE TV_NODEKEY.
    CALL METHOD WA_TREE->GET_SELECTED_NODE
      IMPORTING
        NODE_KEY = STR1.
    IF SY-SUBRC = 0.
      MENUSTR = FCODE.
      NODEKEY = STR1.
      NODEKEYOLD = STR1.

      READ TABLE NODE_TABLE WITH KEY NODE_KEY = STR1 INTO NODE1.
      NODETEXT = NODE1-TEXT.

```

```

        IF FCODE = 'CODE4'.
            CALL METHOD WA_TREE->DELETE_NODE
                EXPORTING
                    NODE_KEY = NODE1-NODE_KEY.
            DELETE NODE_TABLE WHERE NODE_KEY = NODEKEY.
        ENDIF.
    ENDIF.
ENDMETHOD.                                "handle_node_context_menu_sel
ENDCLASS.                                "lcl_application IMPLEMENTATION

*自定义定义对象实例
DATA: G_APPLICATION TYPE REF TO LCL_APPLICATION.

START-OF-SELECTION.
    CALL SCREEN 100.

*&-----*
*&      Module  STATUS_0100  OUTPUT
*&-----*
*      text
*-----*
MODULE STATUS_0100 OUTPUT.
*  SET PF-STATUS 'xxxxxxx'.
*  SET TITLEBAR 'xxx'.
    IF WA_CUSTOM_CONTAINER IS INITIAL.
        CREATE OBJECT G_APPLICATION.
        PERFORM CREATE_TREE.
    ENDIF.
ENDMODULE.                                " STATUS_0100  OUTPUT
*&-----*
*&      Module  USER_COMMAND_0100  INPUT
*&-----*
*      text
*-----*
MODULE USER_COMMAND_0100 INPUT.
    DATA STR1 TYPE TV_NODEKEY.
    SAVE_OK = OK_CODE.
    CLEAR OK_CODE.

    CASE SAVE_OK.
        WHEN 'EXIT'.
            LEAVE PROGRAM.
        WHEN 'GETKEY'.
            CALL METHOD WA_TREE->GET_SELECTED_NODE

```

```

IMPORTING
    NODE_KEY = STR1.

IF SY-SUBRC EQ 0.
    GETKEY = STR1.
ENDIF.
WHEN 'UPTREE'.
    IF ( MENUSTR = 'CODE1' ) OR ( MENUSTR = 'CODE2' ).
        READ TABLE NODE_TABLE
            WITH KEY NODE_KEY = NODEKEY INTO NODE1.
        IF SY-SUBRC EQ 0.
            MESSAGE E208(00) WITH '键值重复! '.
        ENDIF.
    ENDIF.

CASE MENUSTR.
    WHEN 'CODE1'.
        READ TABLE NODE_TABLE WITH KEY NODE_KEY = NODEKEYOLD
            INTO NODE2.

        CLEAR NODE1.
        NODE1-NODE_KEY = NODEKEY.
        NODE1-RELATKEY = NODE2-RELATKEY.
        NODE1-RELATSHIP = CL_GUI_SIMPLE_TREE=>RELAT_LAST_CHILD.
        NODE1-HIDDEN = ''.
        NODE1-DISABLED = ''.
        NODE1-ISFOLDER = 'X'.
        NODE1-N_IMAGE = '@G3@'.
        NODE1-EXP_IMAGE = '@G3@'.
        CLEAR NODE1-EXPANDER.
        NODE1-TEXT = NODETEXT.
        APPEND NODE1 TO NODE_TABLE.
        CLEAR ANODE_TABLE.
        APPEND NODE1 TO ANODE_TABLE.

    CALL METHOD WA_TREE->ADD_NODES
        EXPORTING
            TABLE_STRUCTURE_NAME = 'MTREESNODE'
            NODE_TABLE              = ANODE_TABLE.
        IF SY-SUBRC <> 0.
* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
* WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
        ENDIF.

```

```

        WHEN 'CODE2'.
            CLEAR NODE1.
            NODE1-NODE_KEY = NODEKEY.
            NODE1-RELATKEY = NODEKEYOLD.
            NODE1-RELATSHIP = CL_GUI_SIMPLE_TREE=>RELAT_LAST_CHILD.
            NODE1-HIDDEN = ''.
            NODE1-DISABLED = ''.
            NODE1-ISFOLDER = 'X'.
            NODE1-N_IMAGE = '@G3@'.
            NODE1-EXP_IMAGE = '@G3@'.
            CLEAR NODE1-EXPANDER.
            NODE1-TEXT = NODETEXT.
            APPEND NODE1 TO NODE_TABLE.
            CLEAR ANODE_TABLE.
            APPEND NODE1 TO ANODE_TABLE.

        CALL METHOD WA_TREE->ADD_NODES
            EXPORTING
                TABLE_STRUCTURE_NAME = 'MTREESNODE'
                NODE_TABLE             = ANODE_TABLE.
        IF SY-SUBRC <> 0.
* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
*       WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
        ENDIF.

        WHEN 'CODE3'.
            NODE1-NODE_KEY = NODEKEYOLD.
            CALL METHOD WA_TREE->NODE_SET_TEXT
                EXPORTING
                    NODE_KEY = NODE1-NODE_KEY
                    TEXT      = NODETEXT.
            IF SY-SUBRC <> 0.
* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
*       WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
            ENDIF.

        ENDCASE.
    ENDCASE.
ENDMODULE.
" USER_COMMAND_0100 INPUT
*&-----*
*&      Form  CREATE_TREE
*&-----*
*      text
*-----*
```

```

* --> p1          text
* <-- p2          text
*-----*
FORM CREATE_TREE .
  DATA: EVENTS TYPE CNTL_SIMPLE_EVENTS,
        EVENT1 TYPE CNTL_SIMPLE_EVENT.

  CREATE OBJECT WA_CUSTOM_CONTAINER
    EXPORTING
      CONTAINER_NAME = WA_CONTAINER.

  CREATE OBJECT WA_TREE
    EXPORTING
      PARENT = WA_CUSTOM_CONTAINER
      NODE_SELECTION_MODE = CL_GUI_SIMPLE_TREE=>NODE_SEL_MODE_SINGLE.
  IF SY-SUBRC <> 0.
* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
*       WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
  ENDIF.

  CLEAR NODE1.

  NODE1-NODE_KEY = '国家'.
  CLEAR NODE1-RELATKEY.
  CLEAR NODE1-RELATSHIP.

  NODE1-HIDDEN = ''.
  NODE1-DISABLED = ''.
  NODE1-ISFOLDER = 'X'.

  CLEAR NODE1-N_IMAGE.
  CLEAR NODE1-EXP_IMAGE.
  CLEAR NODE1-EXPANDER.
  NODE1-TEXT = '中国'.
  APPEND NODE1 TO NODE_TABLE.

  CLEAR NODE1.
  NODE1-NODE_KEY = '城市北京'.
  NODE1-RELATKEY = '国家'.
  NODE1-RELATSHIP = CL_GUI_SIMPLE_TREE=>RELAT_LAST_CHILD.
  NODE1-HIDDEN = ''.
  NODE1-DISABLED = ''.
  NODE1-ISFOLDER = 'X'.
  CLEAR NODE1-N_IMAGE.

```

```
CLEAR NODE1-EXP_IMAGE.
CLEAR NODE1-EXPANDER.
NODE1-TEXT = '北京'.
APPEND NODE1 TO NODE_TABLE.
```

```
CLEAR NODE1.
NODE1-NODE_KEY = '城市上海'.
NODE1-RELATKEY = '国家'.
NODE1-RELATSHIP = CL_GUI_SIMPLE_TREE=>RELAT_LAST_CHILD.
NODE1-HIDDEN = ''.
NODE1-DISABLED = ''.
NODE1-ISFOLDER = 'X'.
CLEAR NODE1-N_IMAGE.
CLEAR NODE1-EXP_IMAGE.
CLEAR NODE1-EXPANDER.
NODE1-TEXT = '上海'.
APPEND NODE1 TO NODE_TABLE.
```

```
CLEAR NODE1.
NODE1-NODE_KEY = 'KEY 崇文区'.
NODE1-RELATKEY = '城市北京'.
NODE1-RELATSHIP = CL_GUI_SIMPLE_TREE=>RELAT_LAST_CHILD.
NODE1-HIDDEN = ''.
NODE1-DISABLED = ''.
NODE1-ISFOLDER = ''.
```

*图标, 可运行SHOWICON查阅

```
NODE1-N_IMAGE = '@9Y@'.
NODE1-EXP_IMAGE = '@9Y@'.
CLEAR NODE1-EXPANDER.
NODE1-TEXT = '崇文区'.
APPEND NODE1 TO NODE_TABLE.
```

```
CLEAR NODE1.
NODE1-NODE_KEY = 'KEY 西城区'.
NODE1-RELATKEY = '城市北京'.
NODE1-RELATSHIP = CL_GUI_SIMPLE_TREE=>RELAT_LAST_CHILD.
NODE1-HIDDEN = ''.
NODE1-DISABLED = ''.
NODE1-ISFOLDER = ''.
NODE1-N_IMAGE = '@DF@'.
NODE1-EXP_IMAGE = '@DF@'.
CLEAR NODE1-EXPANDER.
NODE1-TEXT = '西城区'.
APPEND NODE1 TO NODE_TABLE.
```

```

CLEAR NODE1.
NODE1-NODE_KEY = 'KEY 闵行区'.
NODE1-RELATKEY = '城市上海'.
NODE1-RELATSHIP = CL_GUI_SIMPLE_TREE=>RELAT_LAST_CHILD.
NODE1-HIDDEN = ''.
NODE1-DISABLED = ''.
NODE1-ISFOLDER = ''.
NODE1-N_IMAGE = '@NL@'.
NODE1-EXP_IMAGE = '@NL@'.
CLEAR NODE1-EXPANDER.
NODE1-TEXT = '闵行区'.
APPEND NODE1 TO NODE_TABLE.

```

*按照节得内容掏加节得

```

CALL METHOD WA_TREE->ADD_NODES
    EXPORTING
        TABLE_STRUCTURE_NAME      = 'MTREESNODE'
        NODE_TABLE                  = NODE_TABLE
*   EXCEPTIONS
*       ERROR_IN_NODE_TABLE          = 1
*       FAILED                       = 2
*       DP_ERROF                     = 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.

```

```

CALL METHOD WA_TREE->EXPAND_NODE
    EXPORTING
        NODE_KEY      = '国家'
        EXPAND_SUBTREE = 'X'
    EXCEPTIONS
        FAILED          = 1
        ILLEGAL_LEVEL_COUNT = 2
        CNTL_SYSTEM_ERROR = 3
        NODE_NOT_FOUND   = 4
        CANNOT_EXPAND_LEAF = 5.
IF SY-SUBRC NE 0.

ENDIF.

```


***定义双击事件**

```

EVENT1-EVENTID = CL_GUI_SIMPLE_TREE=>EVENTID_NODE_DOUBLE_CLICK.
EVENT1-APPL_EVENT = 'X'.
APPEND EVENT1 TO EVENTS.

EVENT1-EVENTID = CL_GUI_SIMPLE_TREE=>EVENTID_NODE_CONTEXT_MENU_REQ.
EVENT1-APPL_EVENT = ''.
APPEND EVENT1 TO EVENTS.

CALL METHOD WA_TREE->SET_CTX_MENU_SELECT_EVENT_APPL
EXPORTING
    APPL_EVENT = 'X'.
    
```

***添加事件内表**

```

CALL METHOD WA_TREE->SET_REGISTERED_EVENTS
EXPORTING
    EVENTS = 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.
    
```

***将已定义的双击事件分配至树对象**

```

SET HANDLER G_APPLICATION->HANDLE_NODE_DOUBLE_CLICK FOR WA_TREE.

SET HANDLER G_APPLICATION->HANDLE_NODE_CONTEXT_MENU_REQ FOR WA_TREE.

SET HANDLER G_APPLICATION->HANDLE_NODE_CONTEXT_MENU_SEL FOR WA_TREE.
ENDFORM.                " CREATE_TREE
    
```

9.4 Tree Control 对象与 ALV Grid 对象关联

在双击 Tree 对象时, 变更 ALV Trid 对象的显示, 实现界面如图 9-11 所示。

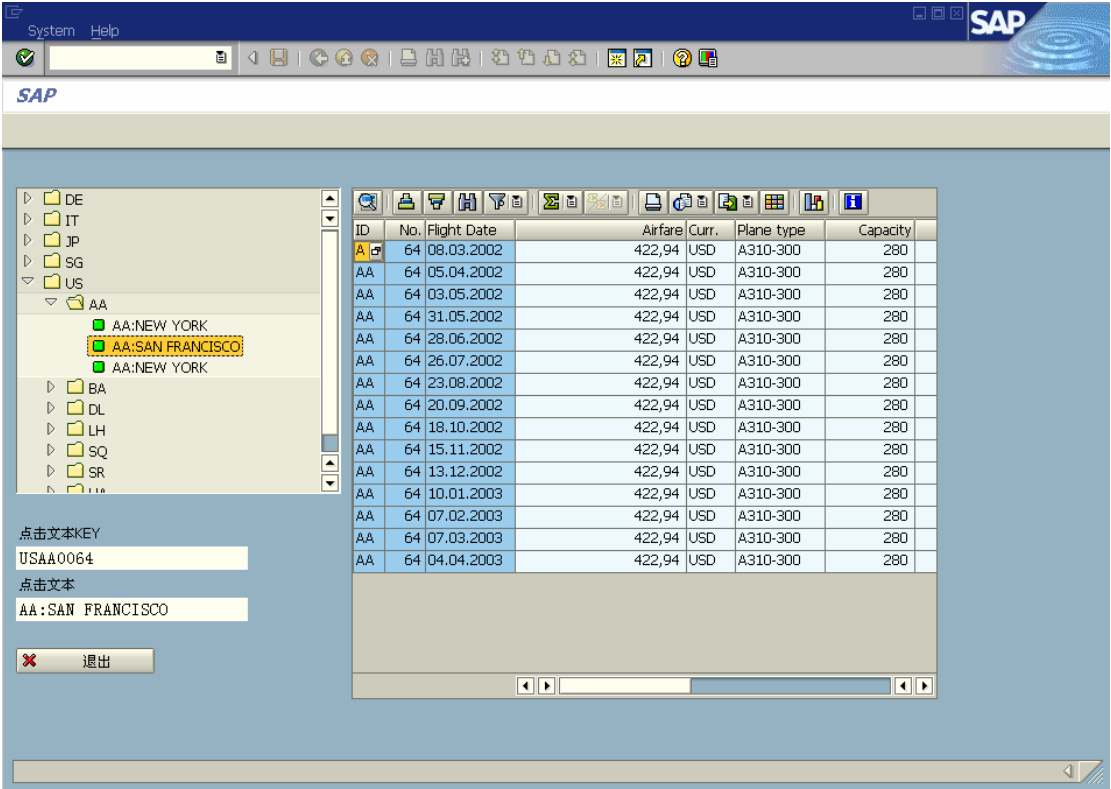


图 9-11

Screen 设计界面如图 9-12 所示。

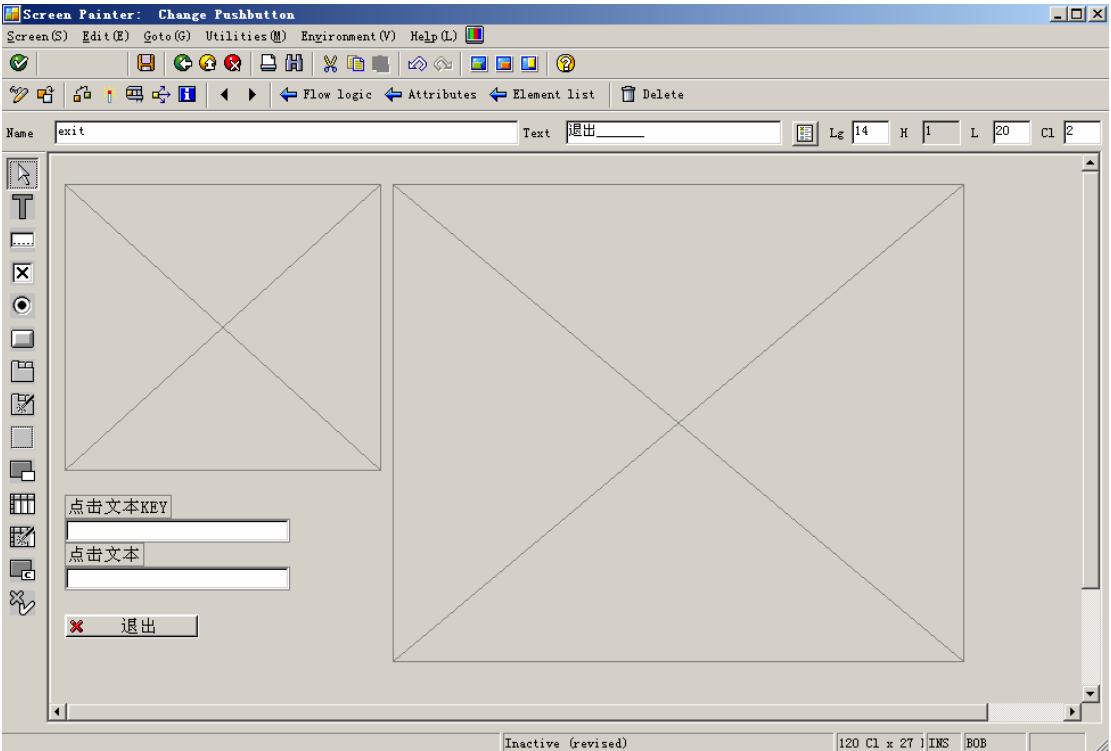


图 9-12

主程序:

```
REPORT YTEST36.
DATA: OK_CODE TYPE SY-UCOMM,
      SAVE_OK TYPE SY-UCOMM.
```

```

DATA: WA_CONTAINER TYPE SCRFNAME VALUE 'TREE1',
      WA_CONTAINERALV TYPE SCRFNAME VALUE 'ALV1',
      WA_CUSTOM_CONTAINER TYPE REF TO CL_GUI_CUSTOM_CONTAINER,
      WA_CUSTOM_CONTAINERALV TYPE REF TO CL_GUI_CUSTOM_CONTAINER,
      WA_TREE TYPE REF TO CL_GUI_SIMPLE_TREE,
      WA_ALV TYPE REF TO CL_GUI_ALV_GRID.

DATA: NODE_TABLE LIKE TABLE OF MTREESNODE,
      NODE1 TYPE MTREESNODE.

DATA WA_SPFLI TYPE TABLE OF SPFLI WITH HEADER LINE.
DATA WA_SFLIGHT TYPE TABLE OF SFLIGHT.

DATA: NODEKEY(200) VALUE 'NODE',
      NODETEXT(200) VALUE 'NODE'.

CLASS LCL_APPLICATION DEFINITION DEFERRED.
DATA EVENT_RECEIVER TYPE REF TO LCL_APPLICATION.
*-----*
*      CLASS lcl_application DEFINITION
*-----*
*
*-----*
CLASS LCL_APPLICATION DEFINITION.
  PUBLIC SECTION.

      METHODS HANDLE_NODE_DOUBLE_CLICK
        FOR EVENT NODE_DOUBLE_CLICK
        OF CL_GUI_SIMPLE_TREE
        IMPORTING NODE_KEY.

ENDCLASS.
"lcl_application DEFINITION
*-----*
*      CLASS lcl_application IMPLEMENTATION
*-----*
*
*-----*
CLASS LCL_APPLICATION IMPLEMENTATION.
  METHOD HANDLE_NODE_DOUBLE_CLICK.
    NODEKEY = NODE_KEY.
    READ TABLE NODE_TABLE WITH KEY NODE_KEY = NODE_KEY
      INTO NODE1.
    NODETEXT = NODE1-TEXT.
  
```

```

CLEAR WA_SFLIGHT.
SELECT * INTO TABLE WA_SFLIGHT FROM SFLIGHT
      WHERE CARRID = NODE1-NODE_KEY+2(2)
      AND CONNID = NODE1-NODE_KEY+4(4).

*   CREATE OBJECT WA_ALV
*       EXPORTING I_PARENT = WA_CUSTOM_CONTAINER.

CALL METHOD WA_ALV->SET_TABLE_FOR_FIRST_DISPLAY
      EXPORTING
        I_STRUCTURE_NAME = 'SFLIGHT'
      CHANGING
        IT_OUTTAB          = WA_SFLIGHT.
ENDMETHOD.                  "handle_node_double_click
ENDCLASS.                   "lcl_application IMPLEMENTATION

DATA G_APPLICATION TYPE REF TO LCL_APPLICATION.

START-OF-SELECTION.
  SELECT * INTO TABLE WA_SPFLI FROM SPFLI
    ORDER BY COUNTRYFR CARRID ASCENDING.
  CREATE OBJECT G_APPLICATION.
  SET SCREEN 100.

*&-----*
*&      Module  STATUS_0100  OUTPUT
*&-----*
*      text
*-----*
MODULE STATUS_0100 OUTPUT.
*  SET PF-STATUS 'xxxxxxxx'.
*  SET TITLEBAR 'xxx'.
  IF WA_CUSTOM_CONTAINER IS INITIAL.
    PERFORM CREATE_TREE.
  ENDIF.

ENDMODULE.                  " STATUS_0100  OUTPUT
*&-----*
*&      Module  USER_COMMAND_0100  INPUT
*&-----*
*      text
*-----*
MODULE USER_COMMAND_0100 INPUT.
  SAVE_OK = OK_CODE.

```

```

    CLEAR OK_CODE.
    CASE SAVE_OK.
        WHEN 'EXIT'.
            LEAVE PROGRAM.
    ENDCASE.
ENDMODULE.                                " USER_COMMAND_0100  INPUT
*&-----*
*&      Form  CREATE_TREE
*&-----*
*      text
*-----*
* -->  p1      text
* <--  p2      text
*-----*
FORM CREATE_TREE .
    DATA: EVENTS TYPE CNTL_SIMPLE_EVENTS,
           EVENT1 TYPE CNTL_SIMPLE_EVENT.
    CREATE OBJECT WA_CUSTOM_CONTAINER
        EXPORTING
            CONTAINER_NAME = WA_CONTAINER.

    CREATE OBJECT WA_CUSTOM_CONTAINERALV
        EXPORTING
            CONTAINER_NAME = WA_CONTAINERALV.

    CREATE OBJECT WA_ALV
        EXPORTING
            I_PARENT = WA_CUSTOM_CONTAINERALV.

    CREATE OBJECT WA_TREE
        EXPORTING
            PARENT = WA_CUSTOM_CONTAINER
            NODE_SELECTION_MODE = CL_GUI_SIMPLE_TREE=>NODE_SEL_MODE_SINGLE.
    IF SY-SUBRC <> 0.
* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
*      WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
    ENDIF.

    DATA: COUNTRYFR LIKE SPFLI-COUNTRYFR,
           CARRID LIKE SPFLI-CARRID.

    DATA: STR1(255).

    LOOP AT WA_SPFLI.

```

```

    IF COUNTRYFR NE WA_SPFLI-COUNTRYFR.
        CLEAR NODE1.
        NODE1-NODE_KEY = WA_SPFLI-COUNTRYFR.
*第一层
        CLEAR NODE1-RELATKEY.
        CLEAR NODE1-RELATSHIP.
*是文件夹
        NODE1-HIDDEN = ''.
        NODE1-DISABLED = ''.
        NODE1-ISFOLDER = 'X'.
*不带ICON
        CLEAR NODE1-N_IMAGE.
        CLEAR NODE1-EXP_IMAGE.
        CLEAR NODE1-EXPANDER.

        NODE1-TEXT = WA_SPFLI-COUNTRYFR.
        APPEND NODE1 TO NODE_TABLE.
    ENDIF.

    IF ( COUNTRYFR NE WA_SPFLI-COUNTRYFR ) OR
        ( CARRID <> WA_SPFLI-CARRID ).
        CLEAR NODE1.
        CONCATENATE WA_SPFLI-COUNTRYFR WA_SPFLI-CARRID
            INTO NODE1-NODE_KEY.
*上层关系
        NODE1-RELATKEY = WA_SPFLI-COUNTRYFR.
        NODE1-RELATSHIP = CL_GUI_SIMPLE_TREE=>RELAT_LAST_CHILD.
        NODE1-HIDDEN = ''.
        NODE1-DISABLED = ''.
        NODE1-ISFOLDER = 'X'.
        CLEAR NODE1-N_IMAGE.
        CLEAR NODE1-EXP_IMAGE.
        CLEAR NODE1-EXPANDER.
*节点显示CARRID
        NODE1-TEXT = WA_SPFLI-CARRID.
        APPEND NODE1 TO NODE_TABLE.
    ENDIF.

    COUNTRYFR = WA_SPFLI-COUNTRYFR.
    CARRID = WA_SPFLI-CARRID.
    CLEAR NODE1.
    CONCATENATE WA_SPFLI-COUNTRYFR WA_SPFLI-CARRID
        WA_SPFLI-CONNID INTO NODE1-NODE_KEY.

```

```

CONCATENATE WA_SPFLI-COUNTRYFR WA_SPFLI-CARRID
    INTO STR1.
NODE1-RELATKEY = STR1.
NODE1-RELATSHIP = CL_GUI_SIMPLE_TREE=>RELAT_LAST_CHILD.
NODE1-HIDDEN = ''.
NODE1-DISABLED = ''.
NODE1-ISFOLDER = 'X'.
NODE1-N_IMAGE = '@5B@'.
NODE1-EXP_IMAGE = '@5B@'.
CLEAR NODE1-EXPANDER.
CONCATENATE WA_SPFLI-CARRID ':' WA_SPFLI-CITYFROM
    INTO STR1.
NODE1-TEXT = STR1.
APPEND NODE1 TO NODE_TABLE.
ENDLOOP.
CALL METHOD WA_TREE->ADD_NODES
    EXPORTING
        TABLE_STRUCTURE_NAME = 'MTREESNODE'
        NODE_TABLE = NODE_TABLE
* EXCEPTIONS
* ERROR_IN_NODE_TABLE = 1
* FAILED = 2
* DP_ERROF = 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.
*定义双击事件
EVENT1-EVENTID = CL_GUI_SIMPLE_TREE=>EVENTID_NODE_DOUBLE_CLICK.
EVENT1-APPL_EVENT = 'X'.
APPEND EVENT1 TO EVENTS.

CALL METHOD WA_TREE->SET_REGISTERED_EVENTS
    EXPORTING
        EVENTS = 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.
*将已定义的双击事件分配至树对象
        SET HANDLER G_APPLICATION->HANDLE_NODE_DOUBLE_CLICK FOR WA_TREE.

ENDFORM.                " CREATE_TREE
    
```