

第六章 实战屏幕 SCREEN 设计

屏幕（Screen）是 ABAP 设计最重要的工作之一，SAP 的单据、主数据维护等业务功能都使用屏幕，一个程序可以包含多个屏幕。

本章主要介绍：

- （1）安装时注意的问题；
- （2）第一个“HELLO WORLD” Screen 程序；
- （3）工具条和菜单设计；
- （4）屏幕对象；
- （5）数据字典关联字段；
- （6）逻辑流；
- （7）下拉框设计；
- （8）修改屏幕；
- （9）子窗口（Subscreen）；
- （10）表条目控制；
- （11）定制控制，上载图片和在屏幕上显示图片实例；
- （12）文本编辑器；
- （13）列表和屏幕相互调用。

6.1 安装时注意的问题

在安装 GUI（客户端）时，需要把屏幕设计的功能选上，如图 6-1 所示。

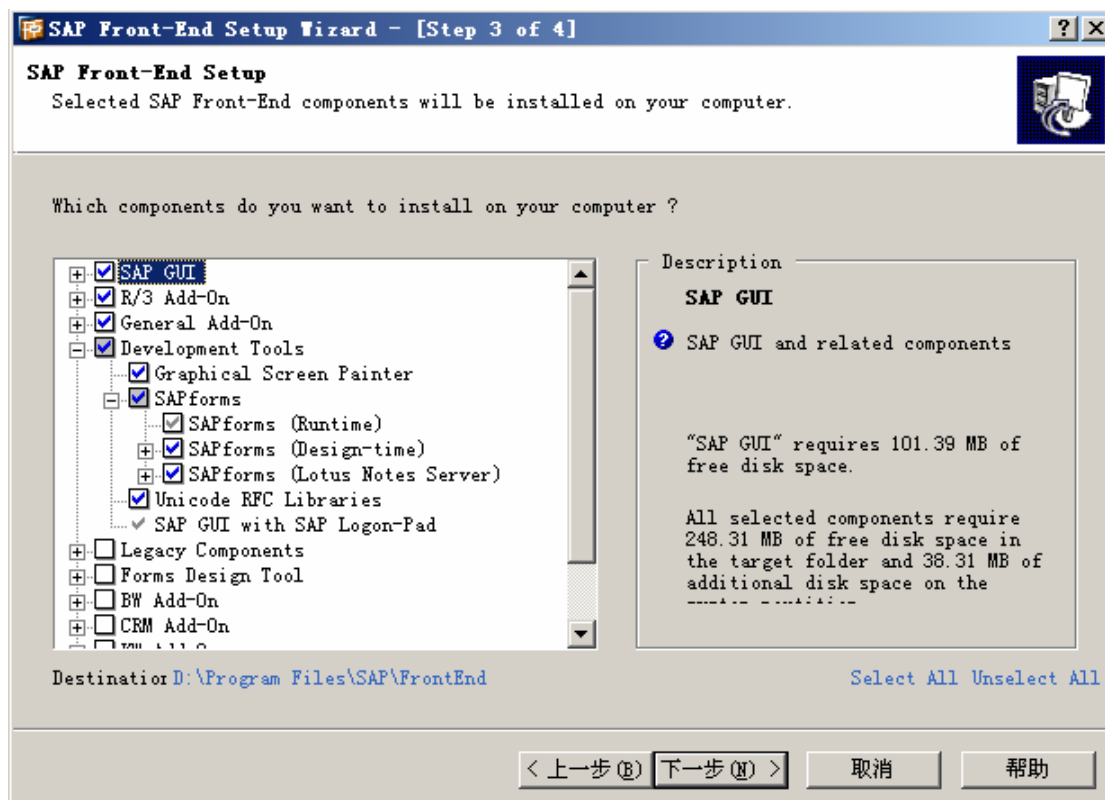


图 6-1

如果没有安装正确，在启动屏幕设计界面时提示“**No response from Graphical Screen Painter-Exiting.**”的错误。

6.2 第一个“HELLO WORLD” Screen 程序

本节将建立一个屏幕，屏幕包含一个“HELLO WORLD!”的文本字段和一个“退出”按钮，单击“退出”按钮时退出整个程序。步骤如下：

- (1) 建立一个新程序；
- (2) 设计 Screen，处理控件、逻辑流；
- (3) 调整程序，调用新建立的 Screen。

6.2.1 建立一个新程序

输入 TCODE：SE38，输入程序名，单击“创建”按钮，如图 6-3 所示。

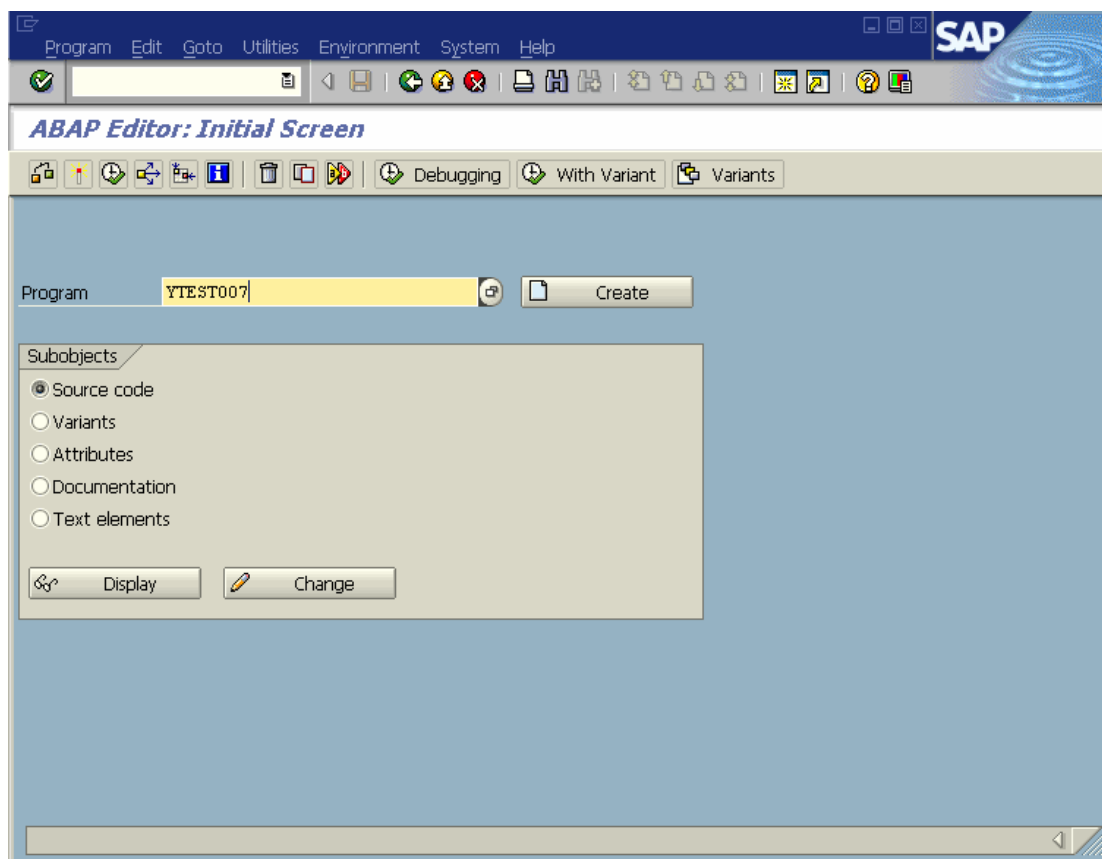


图 6-3

输入标题，程序类型等信息后，单击“保存”按钮，如图 6-4 所示。

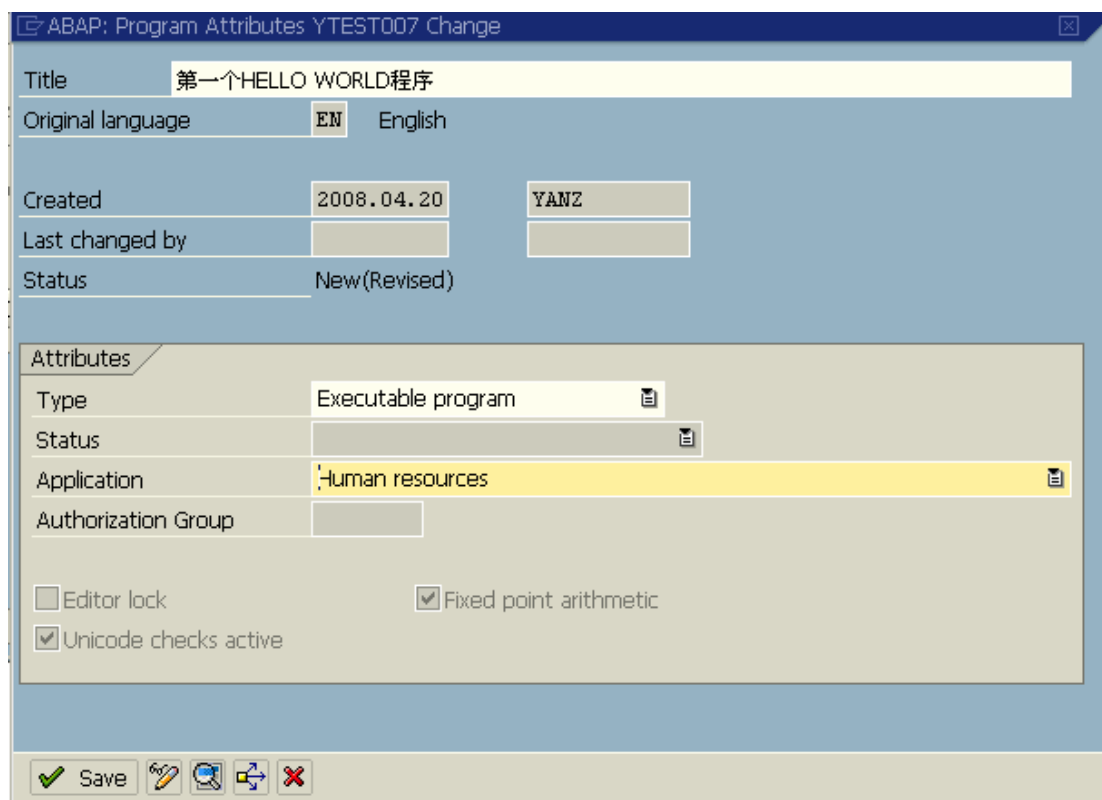


图 6-4

程序不需要传输至其他环境，因而保存时选“本地对象”，如图 6-5 所示。

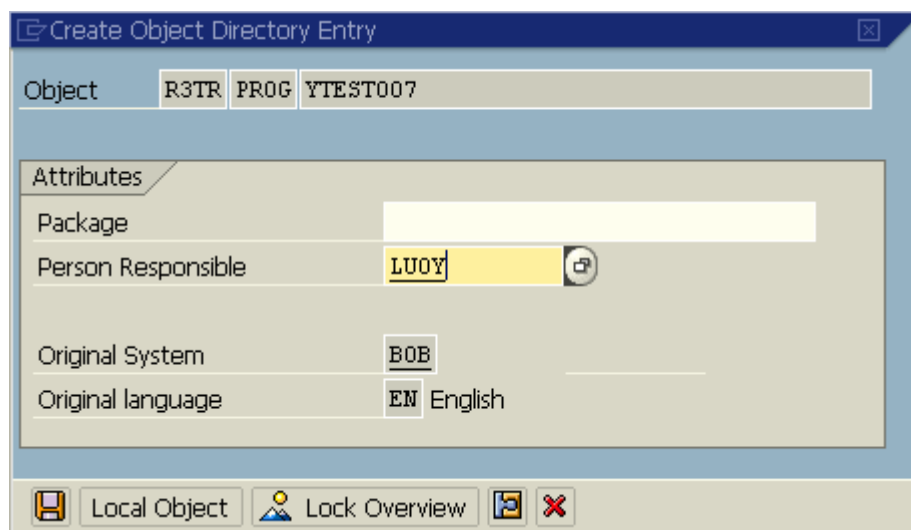


图 6-5

系统进入程序设计界面，直接保存后退回，如图 6-6 所示。

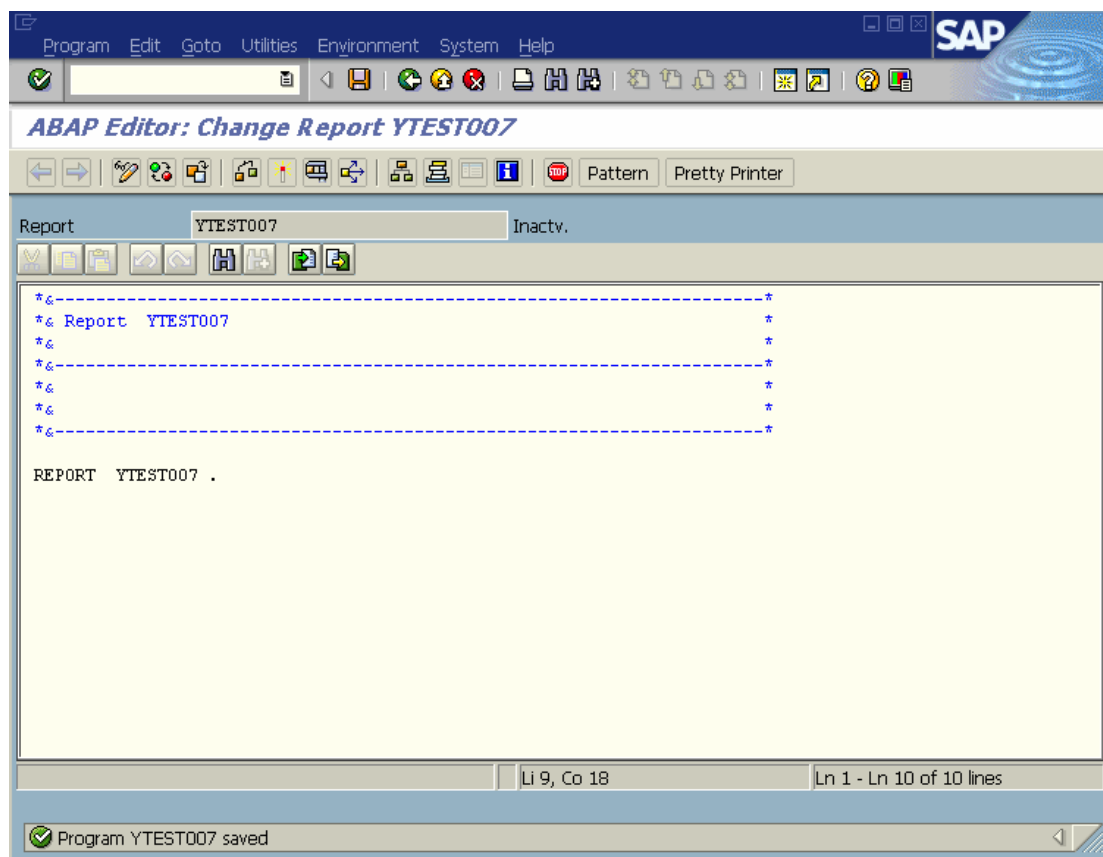


图 6-6

6.2.2 设计 Screen

输入 TCODE: SE51，进入 Screen 设计界面，输入刚建立的程序名“YTEST007”，输入屏幕编号 100 后单击“创建”按钮，如图 6-7 所示。

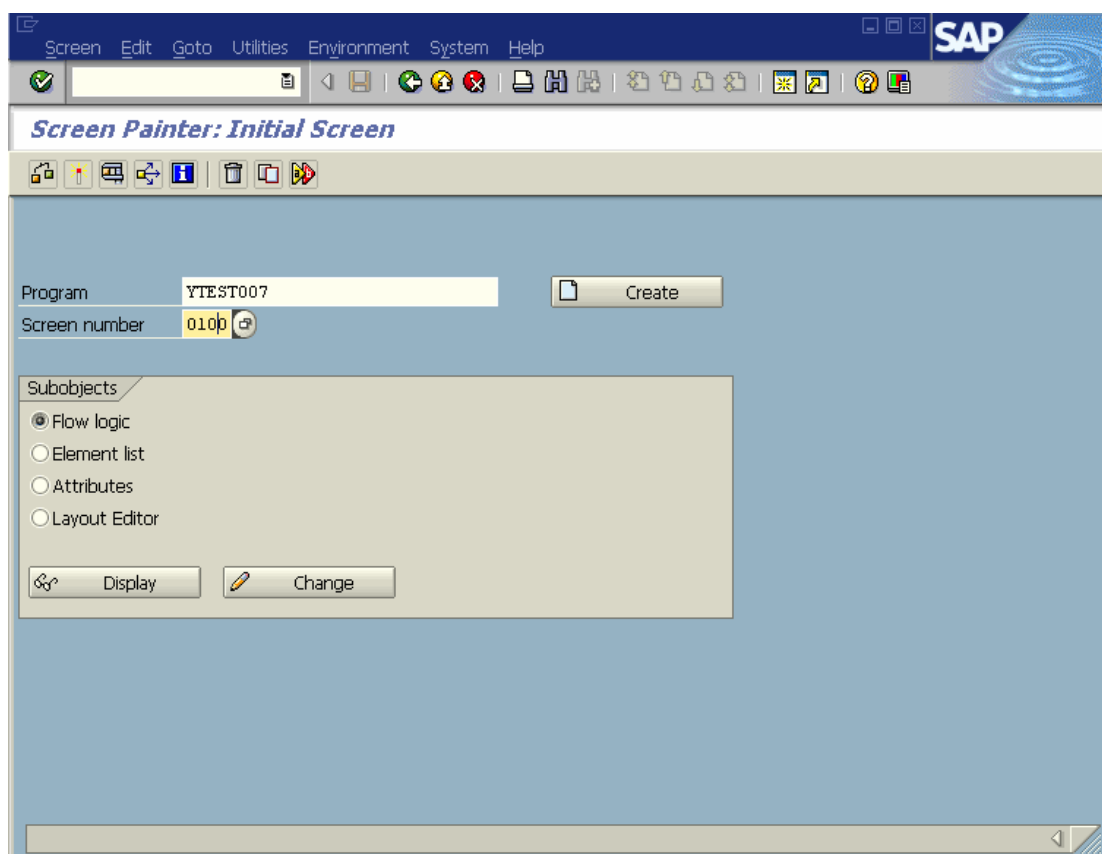


图 6-7

在属性页输入描述后，直接保存，如图 6-8 所示。

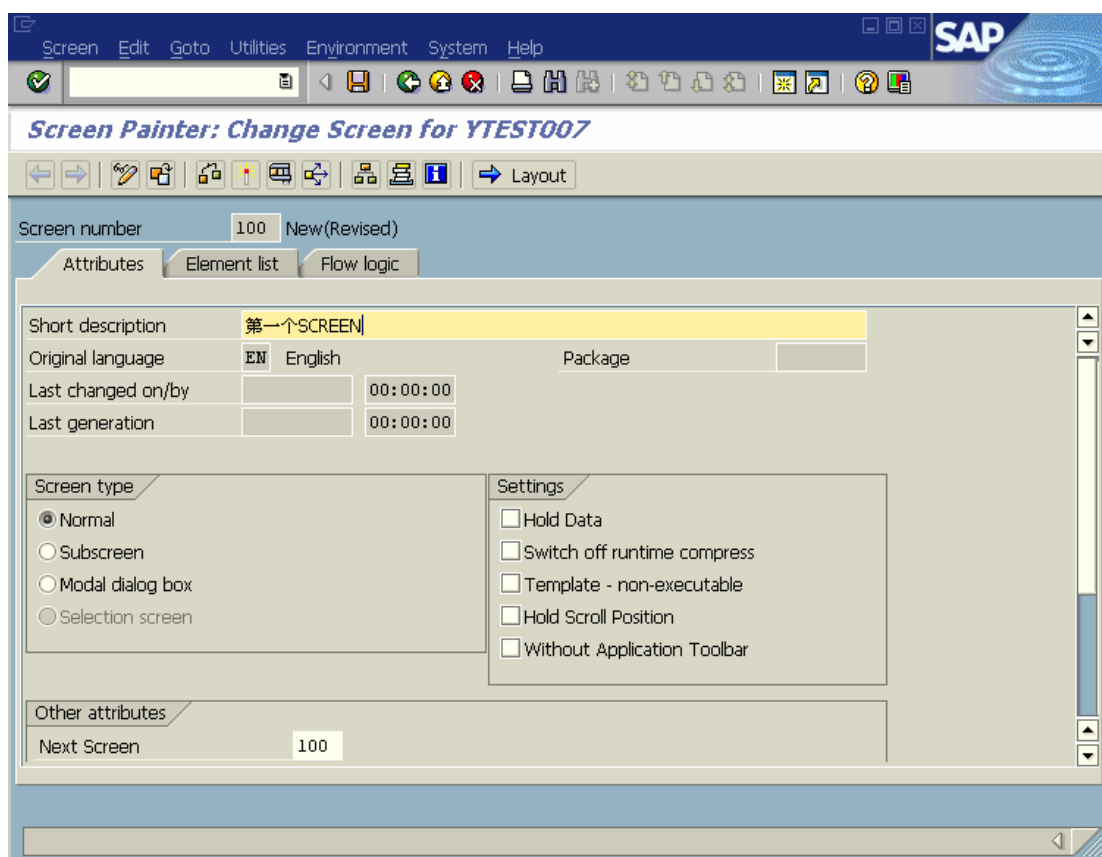


图 6-8

单击工具条上的“格式”按钮，进入设计界面，添加一个文本字段控件，输入名称和文本，如果控件显示红色的表示未正确设置属性，如图 6-9 所示。

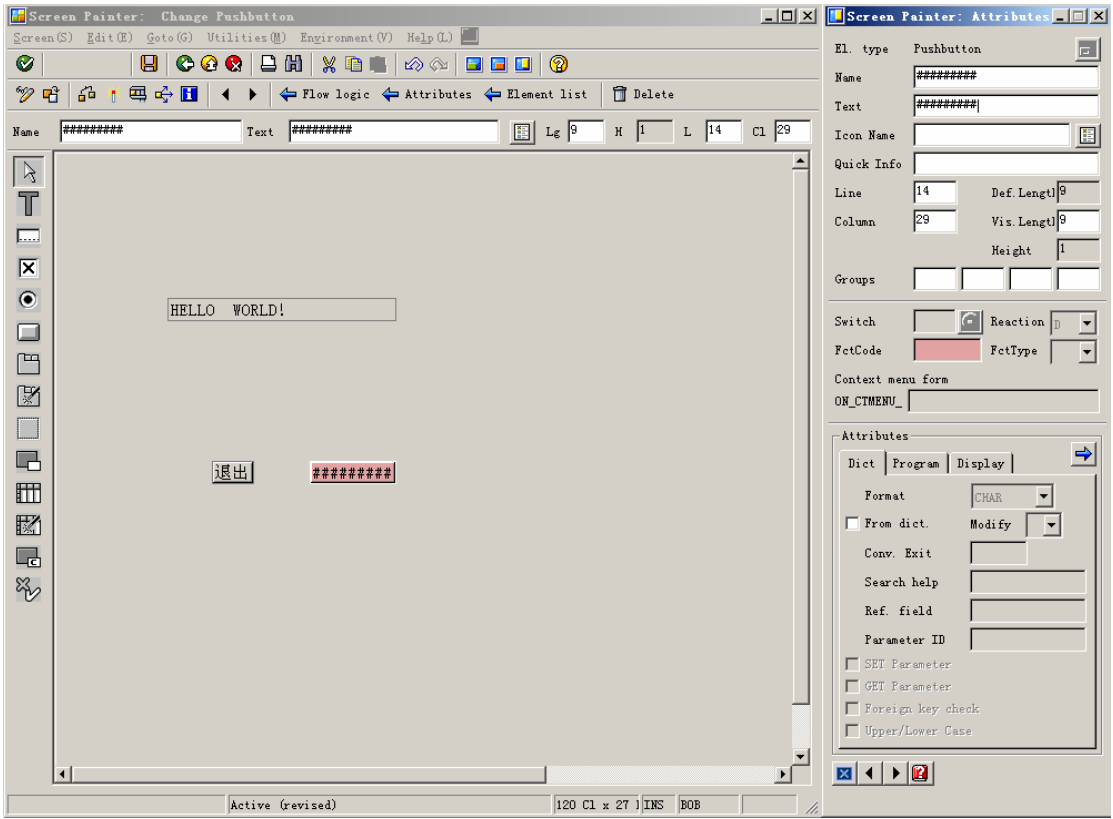


图 6-9

“退出”按钮属性如图 6-10 所示。

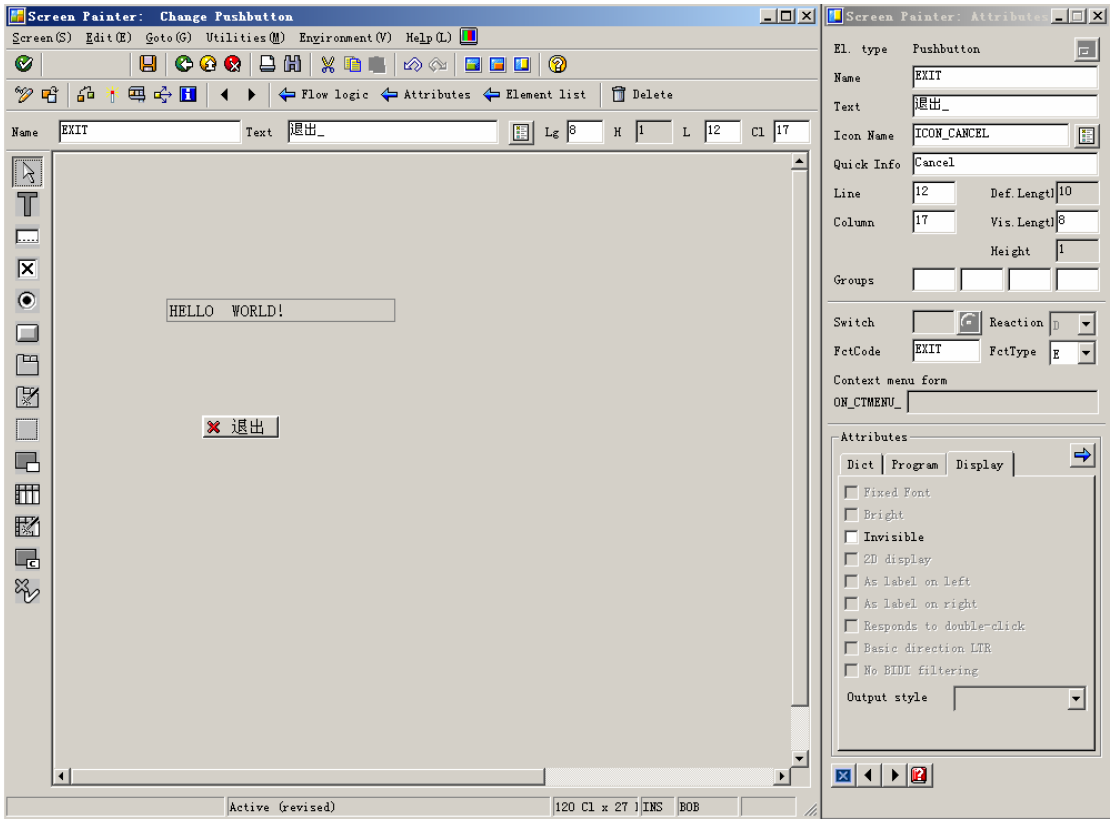


图 6-10

注意属性中功能码为 EXIT，函数类型“E”，表示退出函数。
设计窗口上的按钮说明：

- 文本元素
- 输入框
- 复选框
- 单选按钮
- 按钮
- 多页表条目控制制作
- 表条目控制制作向导
- 对象面板
- 子窗口、包含其他窗口
- 表格
- 表格制作导航
- 其他对象
- 状态图标

Flow logic 控制用屏幕输入输出事件，如在启动屏幕时使用的菜单、工具条，在输入后

对输入框的检查，对按钮事件的控制等。

← Attributes 窗口的属性。

← Element list 窗口上所有对象的列表，属性等。

单击 ← Flow logic，出现如图 6-11 所示窗口。

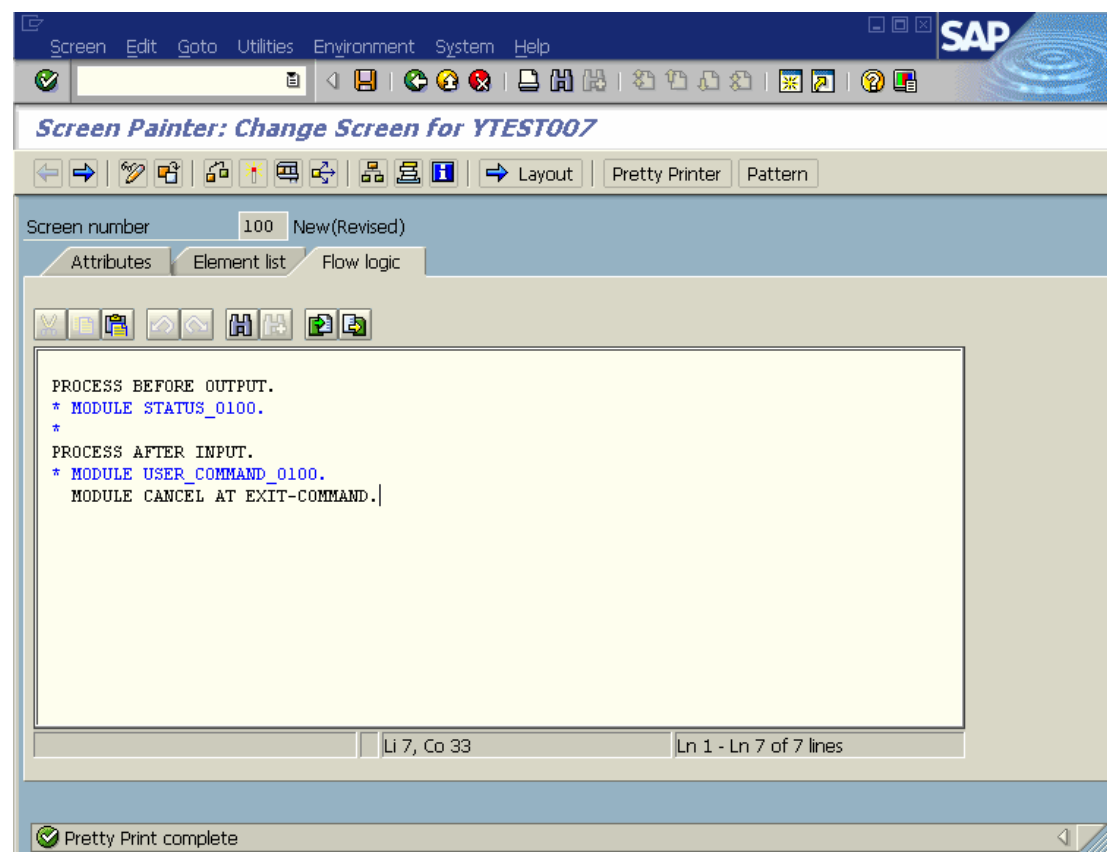


图 6-11

PROCESS BEFORE OUTPUT(PBO): 屏幕显示前的调用模块。

PROCESS AFTER INPUT(PAI): 响应用户输入后的调用模块，如单击按钮，输入字段后回车等事件。

写入代码“MODULE CANCEL AT EXIT-COMMAND.”。表示在退出事件时执行 CANCEL 模块，这和退出按钮的函数类型属性对应。

6.2.3 从程序中调用 Screen

编辑程序 YTEST007，调整后程序：

REPORT YTEST007 .

*直接调用窗口

CALL SCREEN 100.

```
*-----*
```

```
*  MODULE cancel INPUT
```

```
*-----*
```


*

*CANCEL MODULE与屏幕逻辑流对应

MODULE CANCEL INPUT.

LEAVE PROGRAM.

ENDMODULE. "cancel INPUT

执行程序，输出结果如图 6-12 所示。

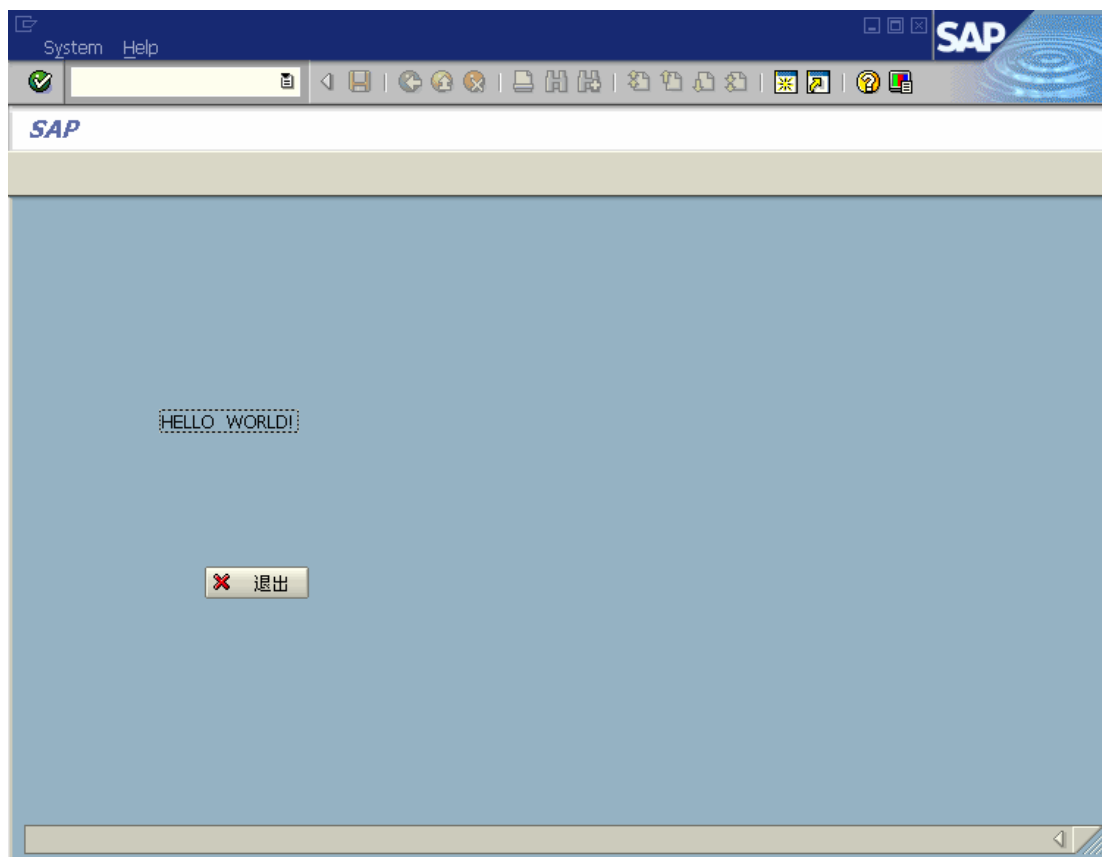


图 6-12

单击“退出”按钮，退出程序。

6.3 工具条和菜单设计

在“HELLO WORLD!”程序基础上，在屏幕上建立菜单，工具条按钮、系统按钮、添加编辑框；在程序启动时，将程序中的变量传递到窗口的编辑框中；在单击菜单、工具条按钮、系统按钮时，将选中对象的功能码传递到文本框；离开屏幕时，将两个编辑框内容输出，表示窗口变量正确传递回主程序。

学习本章后，会对屏幕布局及相关控制有一个完整的了解，并且有融会贯通的感觉，所以本节非常重要，主要介绍：菜单编辑器，应用工具条设计，菜单设计，系统按钮设计，逻辑流程设计，加班编辑框，加入 OK_CODE，代码设计，输出结果。

6.3.1 菜单编辑器

工具条和菜单设计，需要用菜单编辑器功能（TCODE: SE41），可以在屏幕设计的通过单击工具条按钮进入。单击工具条的“其他对象”按钮，如图 6-13 所示。

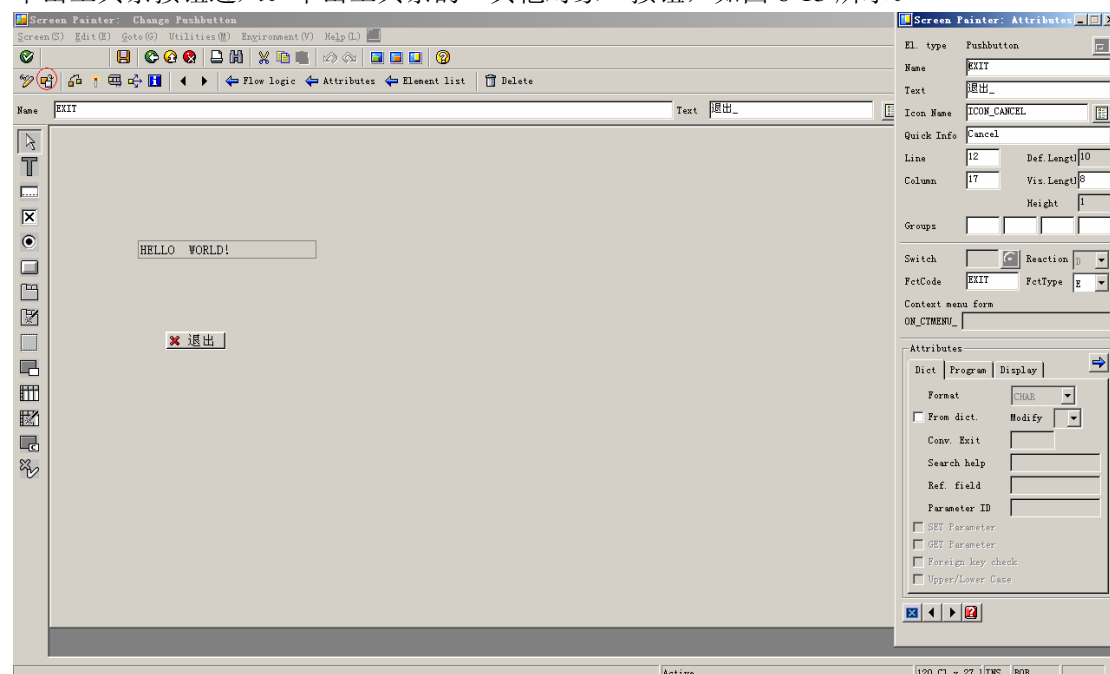


图 6-13

在“GUI”状态名称后，单击创建按钮（），如图 6-14 所示。

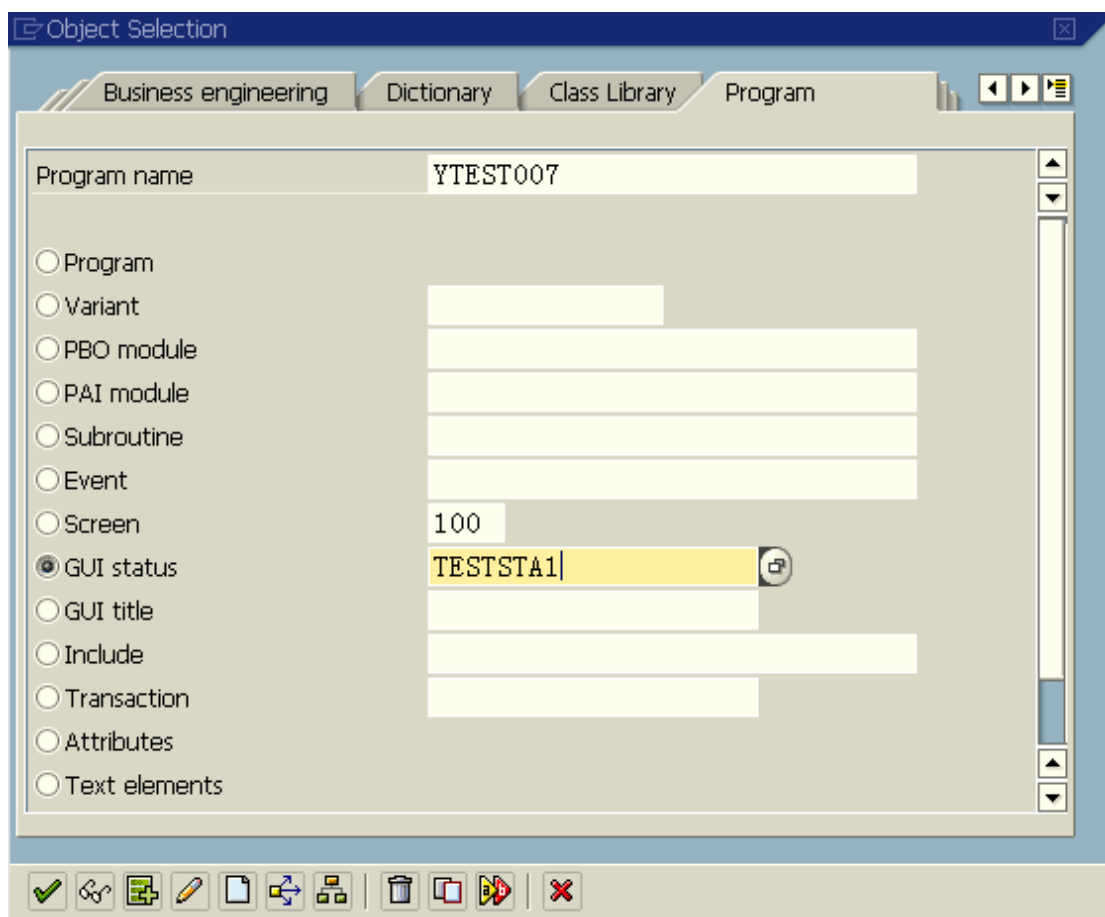


图 6-14

输入短文本，如图 6-15 所示。



图 6-15

6.3.2 应用工具条设计

展开“应用工具条”，在项目 1 输入“BIN1”，并双击它，如图 6-16 所示。

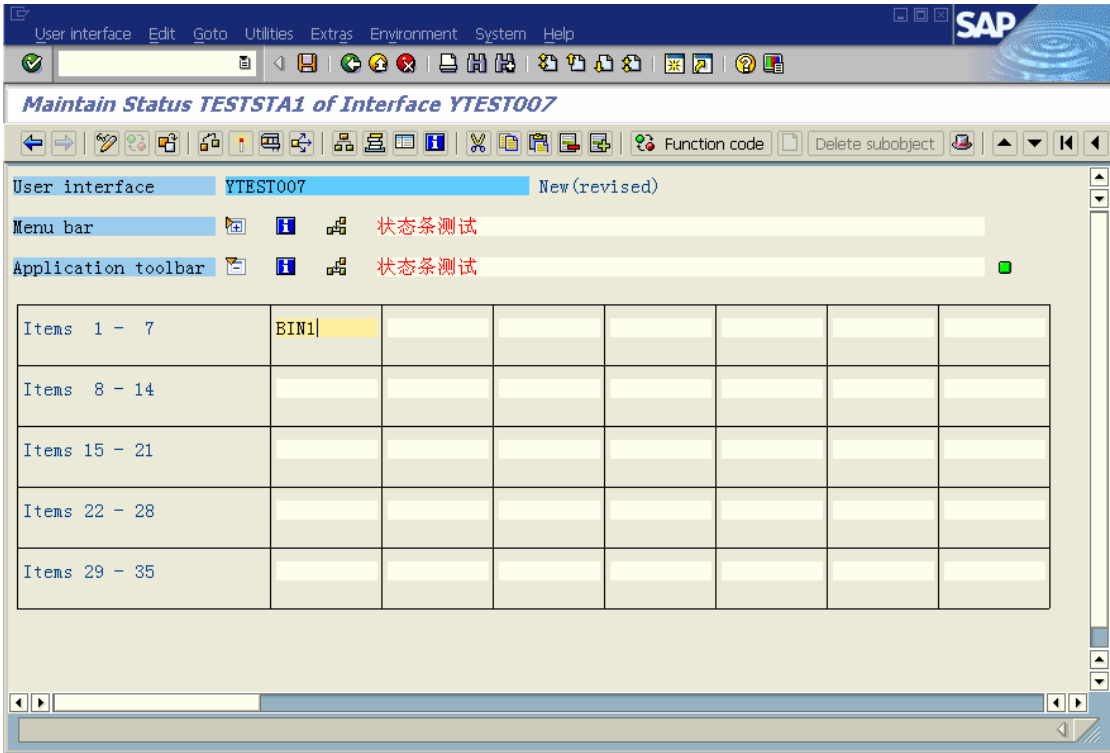


图 6-16

选“静态文本”，继续，如图 6-17 所示。

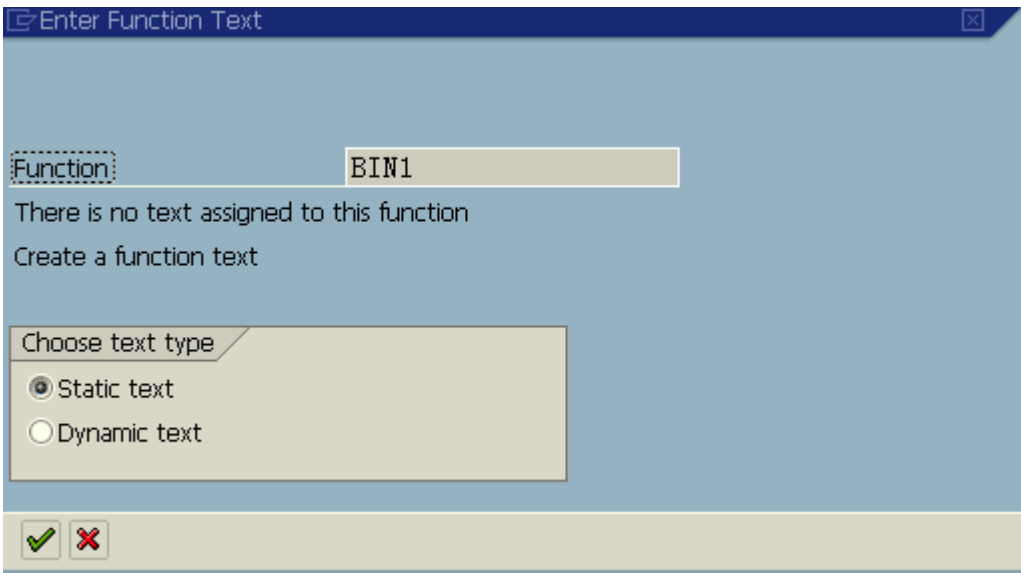


图 6-17

输入函数文本，可以选择图标，但在选择图标后，该按钮仅显示图标，文本做提示，如图 6-18 所示。

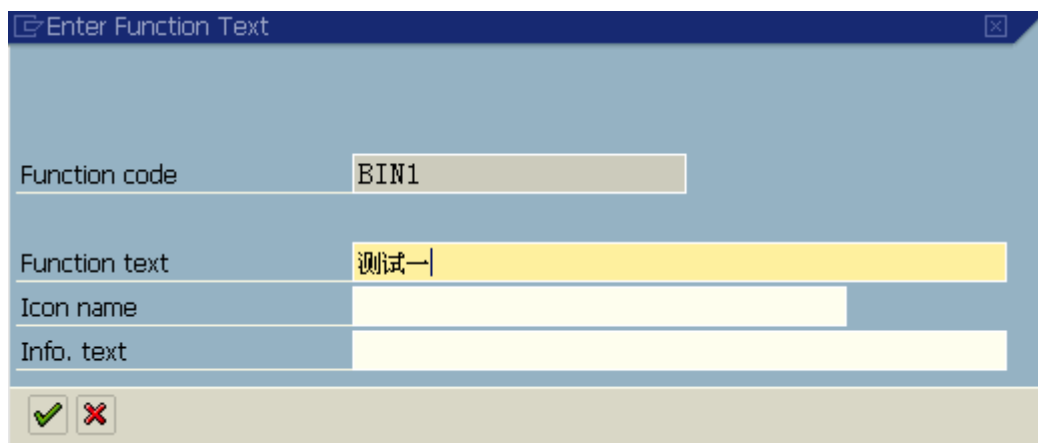


图 6-18

继续，为按钮分配一个功能键，如图 6-19 所示。

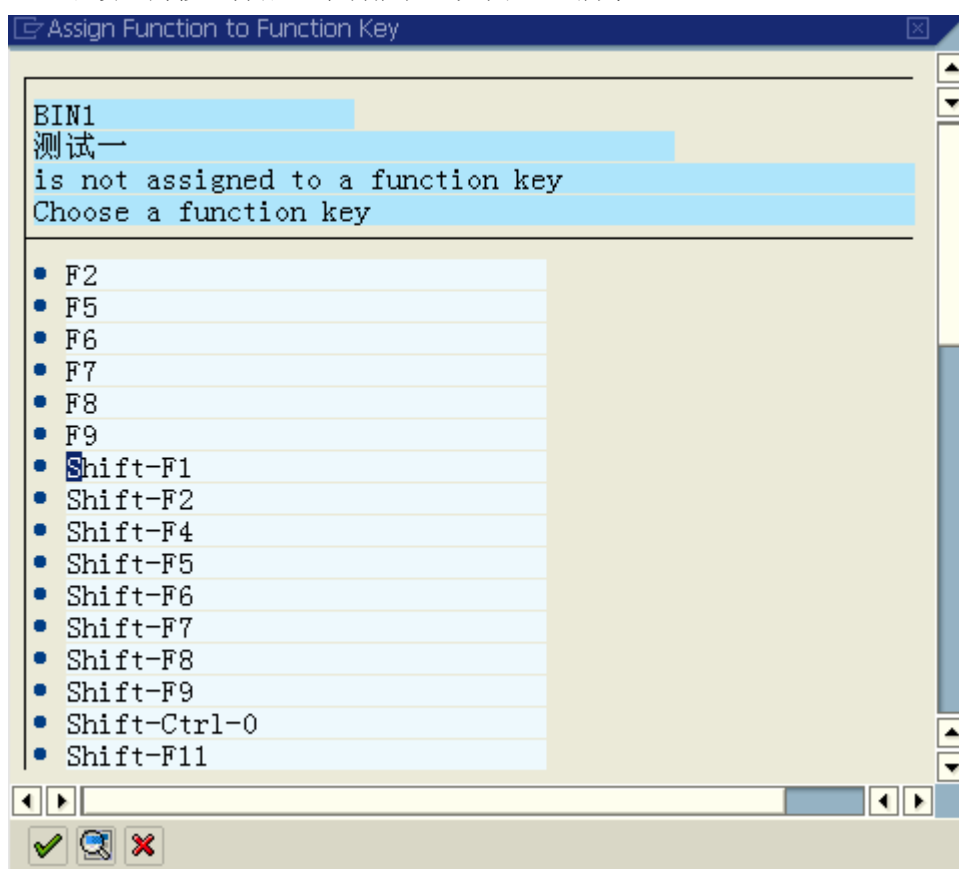


图 6-19

输入图标文字，如图 6-20 所示。

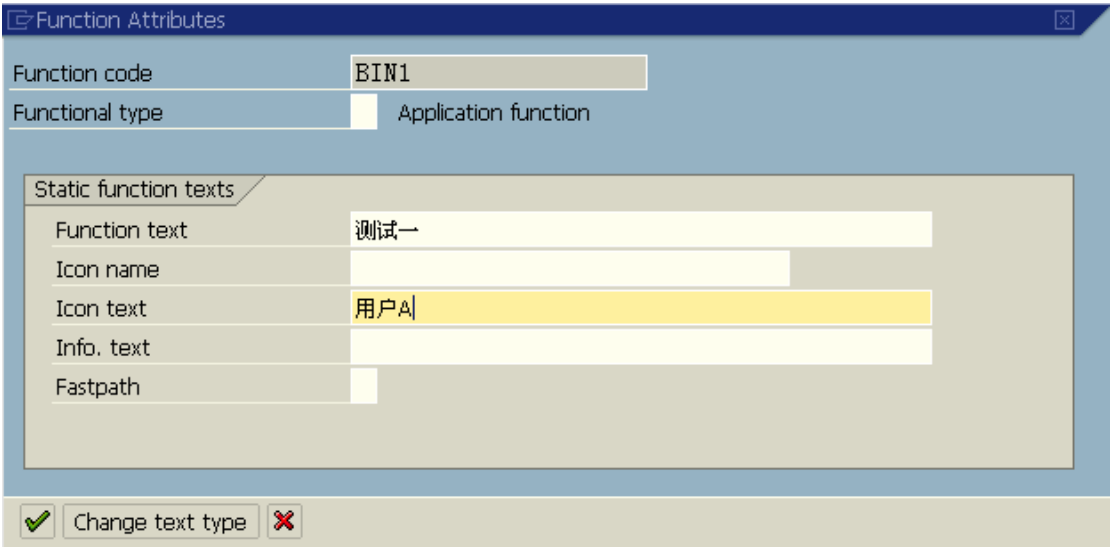


图 6-20

以同样的方式添加第 2 个按钮，如图 6-21 所示。

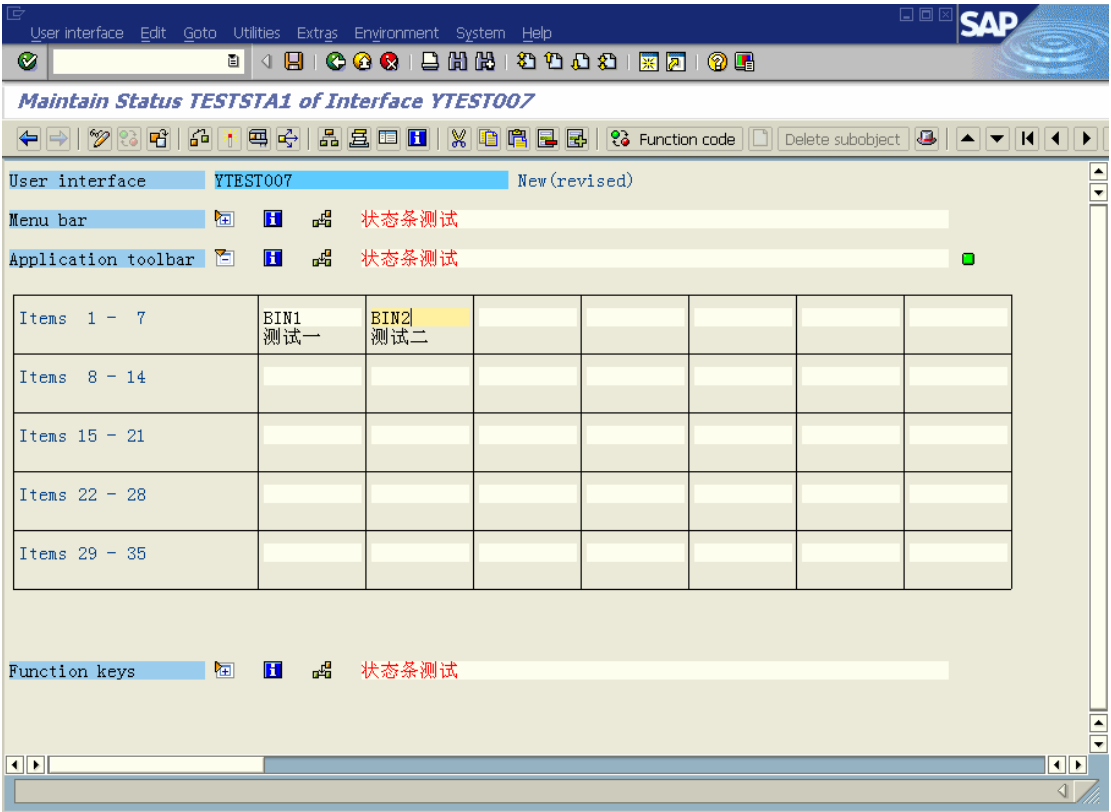



图 6-21

完成后单击工具条上的按钮，输出结果如图 6-22 所示。

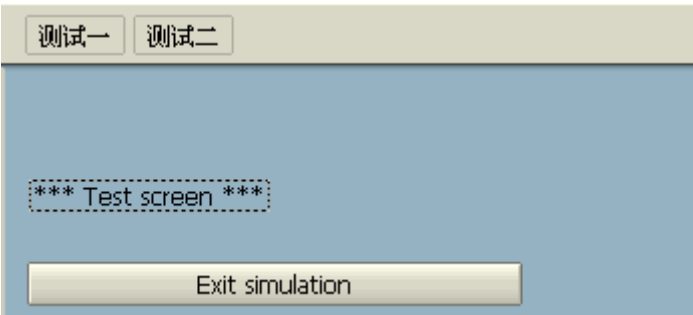


图 6-22

6.3.3 菜单设计

单击“菜单栏”，展开它，输入主菜单后双击它，输入相应的菜单项，如图 6-23 所示。

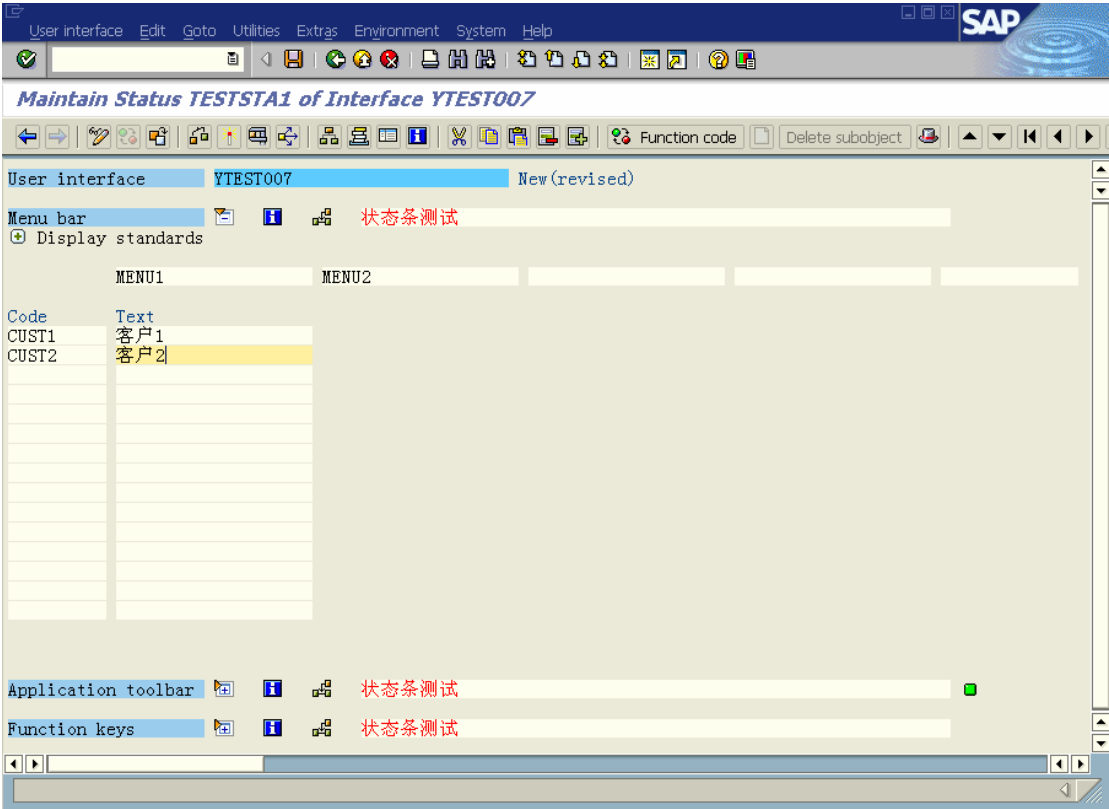


图 6-23

单击测试按钮，输出界面如图 6-24 所示。

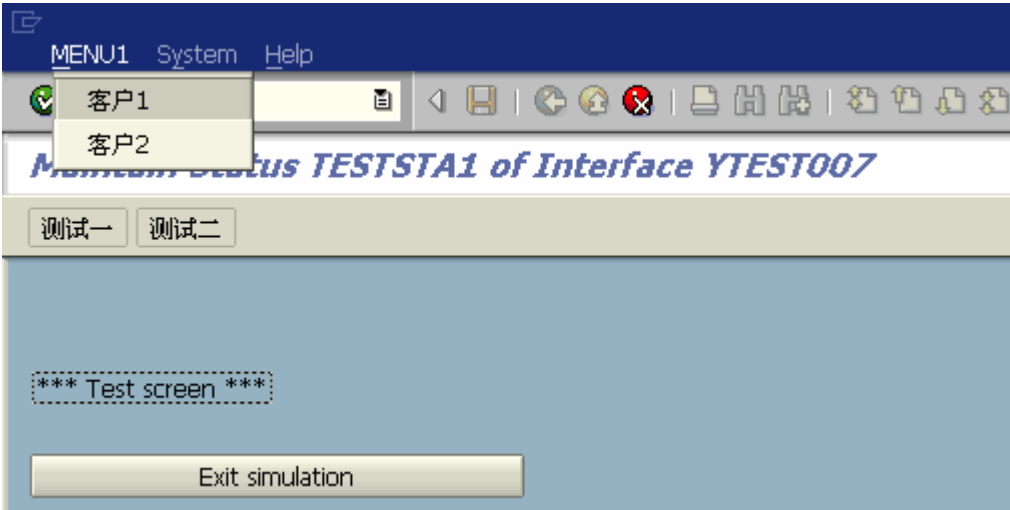


图 6-24

6.3.4 系统按钮设计

与菜单和工具条按钮设计一样，单击“功能键”，如图 6-25 所示。

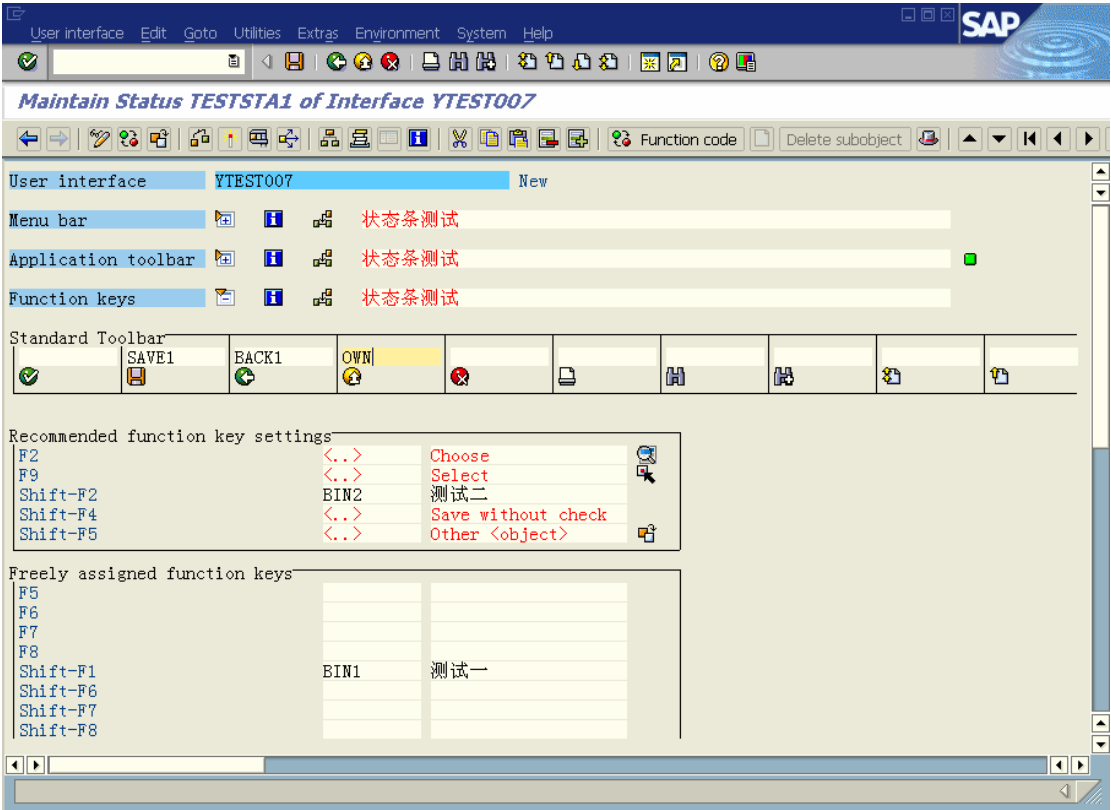





图 6-25

在标准工具条输入功能代码，测试界面如图 6-26 所示。



图 6-26

设计完成保存后，单击工具条上的按钮，激活状态条。

单击按钮回到 Screen 设计界面，单击按钮设计逻辑流。

6.3.5 逻辑流设计

将默认的 PAI 和 PBO 两个 MODULE 设为有效（去掉*注释），如图 6-27 所示。

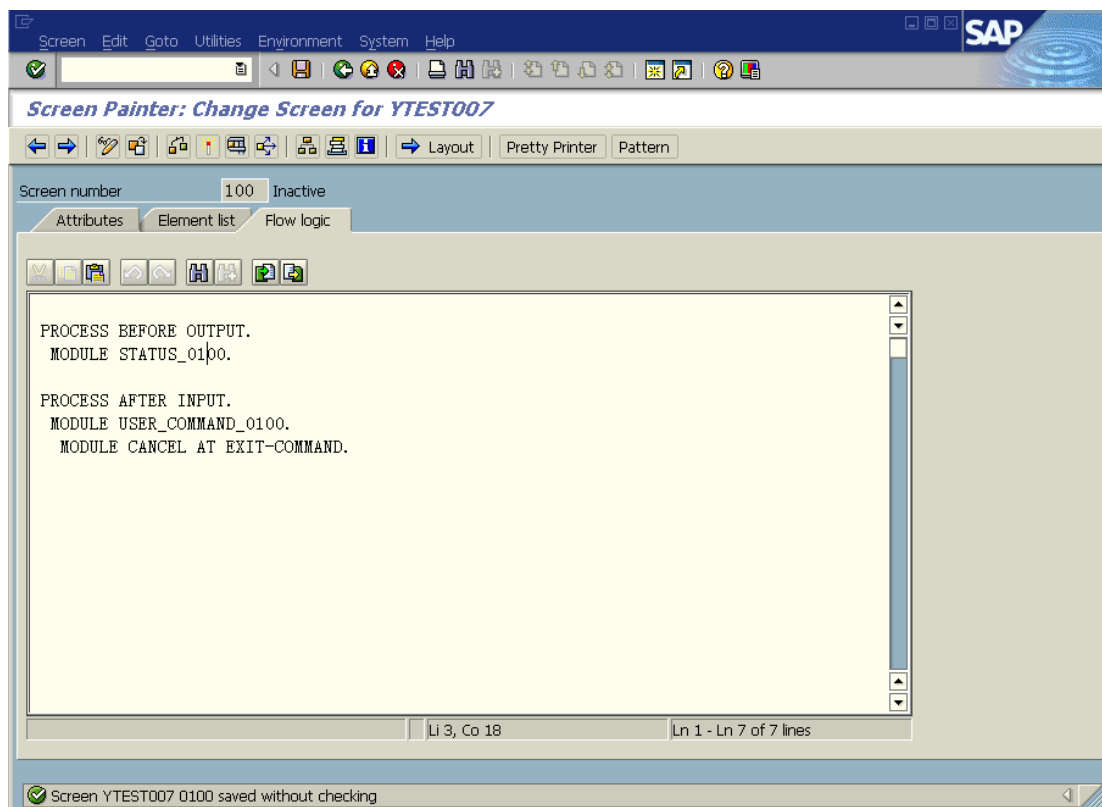


图 6-27

在 MODULE STATUS_0100 调用设计的状态条,在 MODULE USER_COMMAND_0100 中处理各类按钮事件。

6.3.6 输入字段

进入屏幕设计界面，添加两个文本字段和两个输入字段。选择输入字段时，设定属性为输入输出字段，如图 6-28 所示。

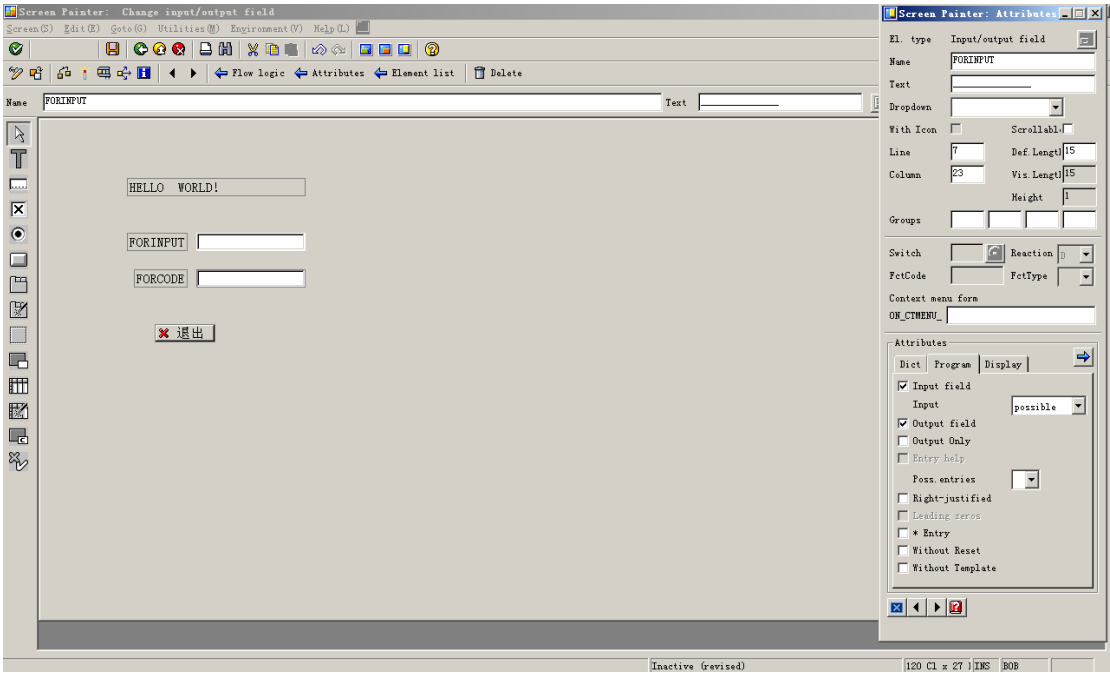


图 6-28

6.3.7 OKCODE

单击  Element list 按钮，定义 OK_CODE 变量来处理屏幕交互，如图 6-29 所示。

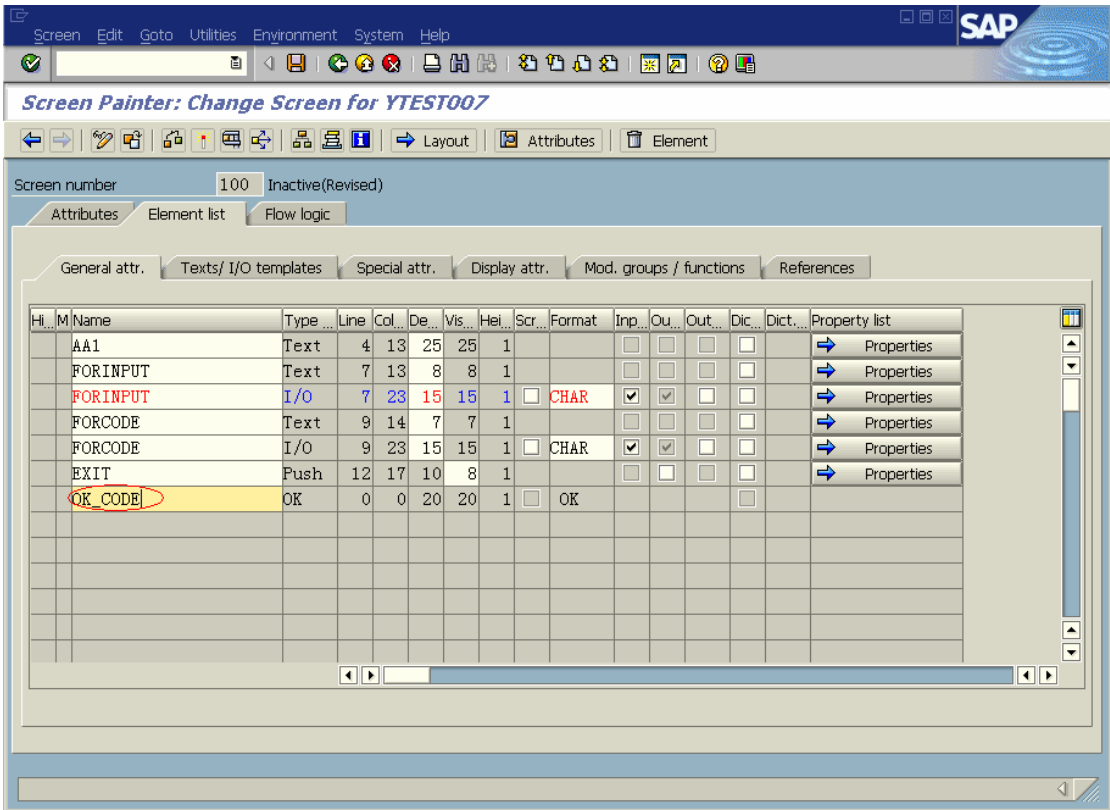


图 6-29

保存并激活该屏幕。

6.3.8 程序设计

```
REPORT  YTEST007 .

DATA: FORINPUT(20) TYPE C VALUE '初始值',
      FORCODE(20)  TYPE C.
*功能码返回值
DATA: OK_CODE TYPE SY-UCOMM,
      SAVE_OK  TYPE SY-UCOMM.

*直接调用窗口
CALL SCREEN 100.
*输出返回变量
WRITE: FORINPUT, / FORCODE.
*-----*
*  MODULE status_0100 OUTPUT
*-----*
*  PBO输入前控制
*-----*
MODULE STATUS_0100 OUTPUT.
*定义状态条、包括菜单、工具条按钮、系统按钮等
  SET PF-STATUS 'TESTSTA1'.
*将变量值输出至屏幕字段
  FORCODE = SAVE_OK.
ENDMODULE.                                "STATUS_0100 OUTPUT
*-----*
*  MODULE cancel INPUT
*-----*
*  CANCEL MODULE与屏幕逻辑流对应
*-----*
MODULE CANCEL INPUT.
  LEAVE PROGRAM.
ENDMODULE.                                "cancel INPUT
*-----*
*  MODULE user_command_0100 INPUT
*-----*
*  PAI输入后控制
*-----*
MODULE USER_COMMAND_0100 INPUT.
  SAVE_OK = OK_CODE.
  CLEAR OK_CODE.
```

*分析功能码, 如果是工具条上的第2个按钮, 则退出当前屏幕

```
CASE SAVE_OK.
  WHEN 'BIN2' .
    LEAVE TO SCREEN 0.
ENDCASE.
ENDMODULE.                                "user_command_0100 INPUT
```

6.3.9 屏幕输出

屏幕输出如图 6-30 所示。

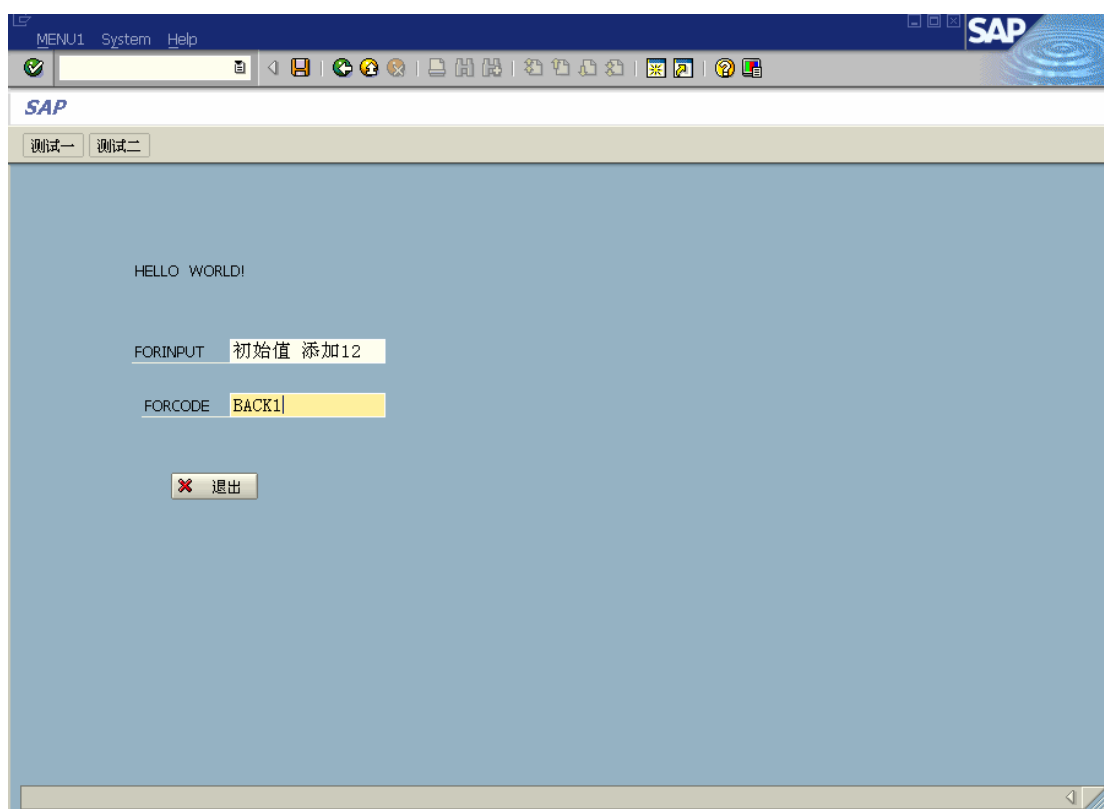


图 6-30

单击“测试二”按钮，退出设计屏幕，系统输出输入框内容如图 6-31 所示。


```
初始值 添加12
BACK1
```

(由于我做的时候没有安装中文，所以会出现乱码的情况)

图 6-31

6.4 屏幕对象功能

6.4.1 单选按钮组的定义

单选按钮组是多个单选按钮（）的组合操作。选中多个单选按钮，单击右键选菜单“单选按钮组”→“定义”，如图 6-32 所示。

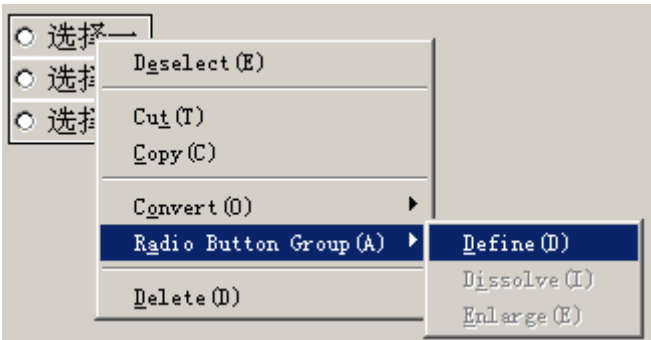


图 6-32

取消多个单选按钮组合单击“分解”菜单。

6.4.2 输入输出字段的属性

输入输出字段的属性如图 6-33 所示。

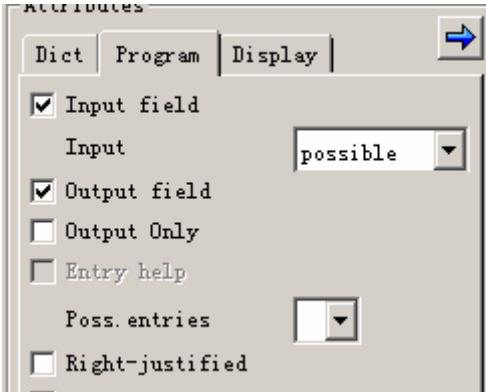
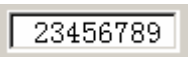


图 6-33

“输出字段” 

“输入输出” 

“强制性输入” 

“只输出” 

“不可见”设计界面

6.4.3 数据字典关联字段

在 Screen 设计过程中，数据字典关联字段可以根据数据字典中定义的透明表、结构等信息，方便地在屏幕上建立与字典关联的文本、输入输出字段。

例如使用航班计划表（SPFLI），在屏幕上定义相关该表字段，并根据输入的主关键字检索其他字段的信息，并显示到输入输出字段。

航班计划表结构，如图 6-34 所示。

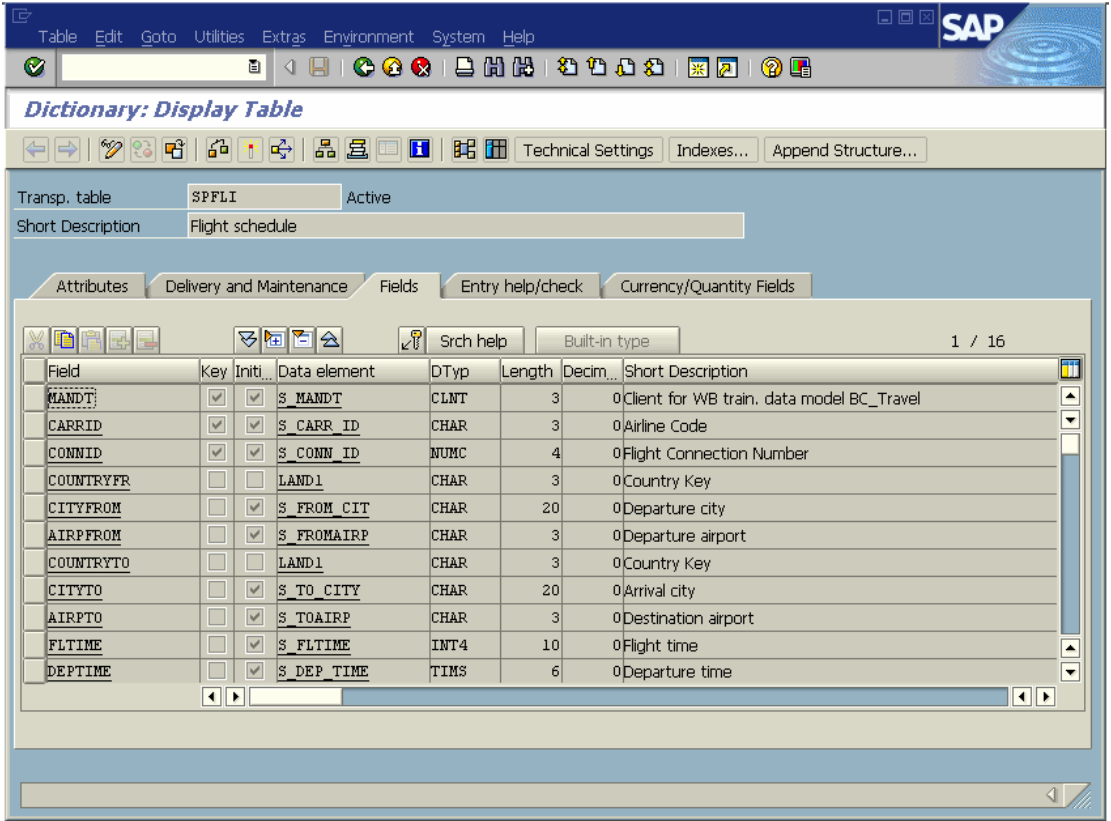



图 6-34

MANDT 字段表示集团，在登录时已默认，本节通过 CARRID 和 CONNID 两个关键字检索数据。

新建一个程序并建立一个屏幕后，单击图 6-35 中所示的“字典字段窗口”按钮 ()。

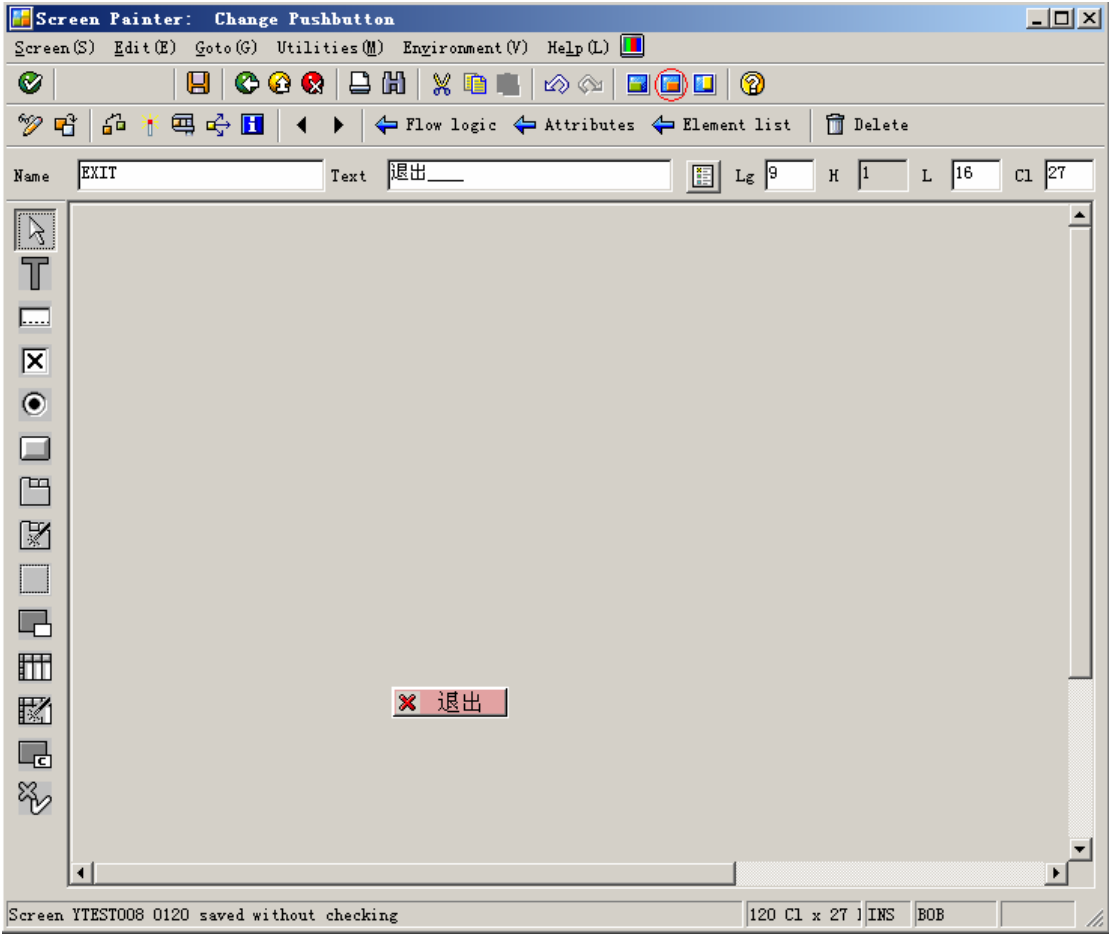


图 6-35

输入表“SPFLI”后，回车，系统显示该表所有字段，选择需要输出的字段，如图 6-36 所示。

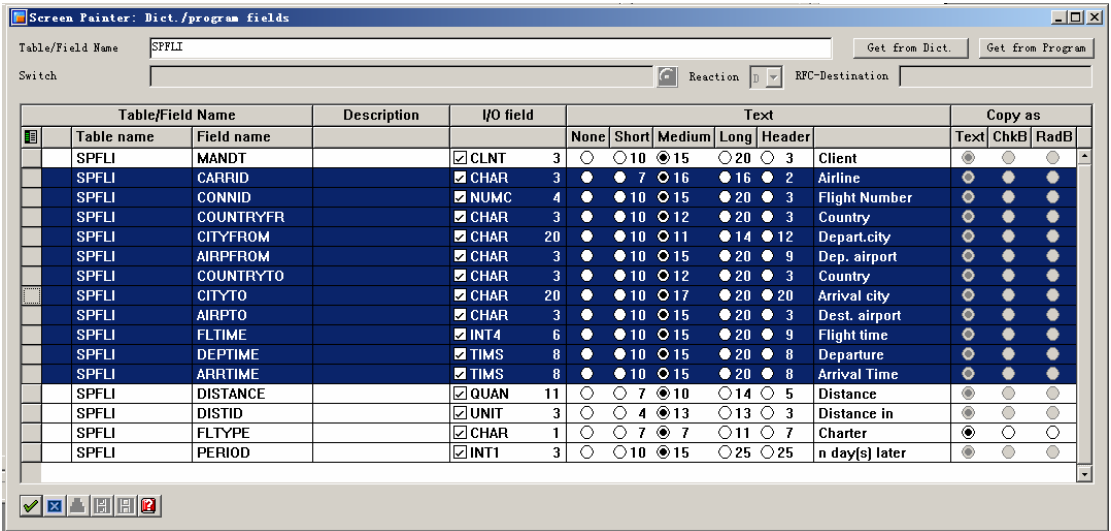


图 6-36

定位输出位置后，所有字段自动输出，如图 6-37 所示。

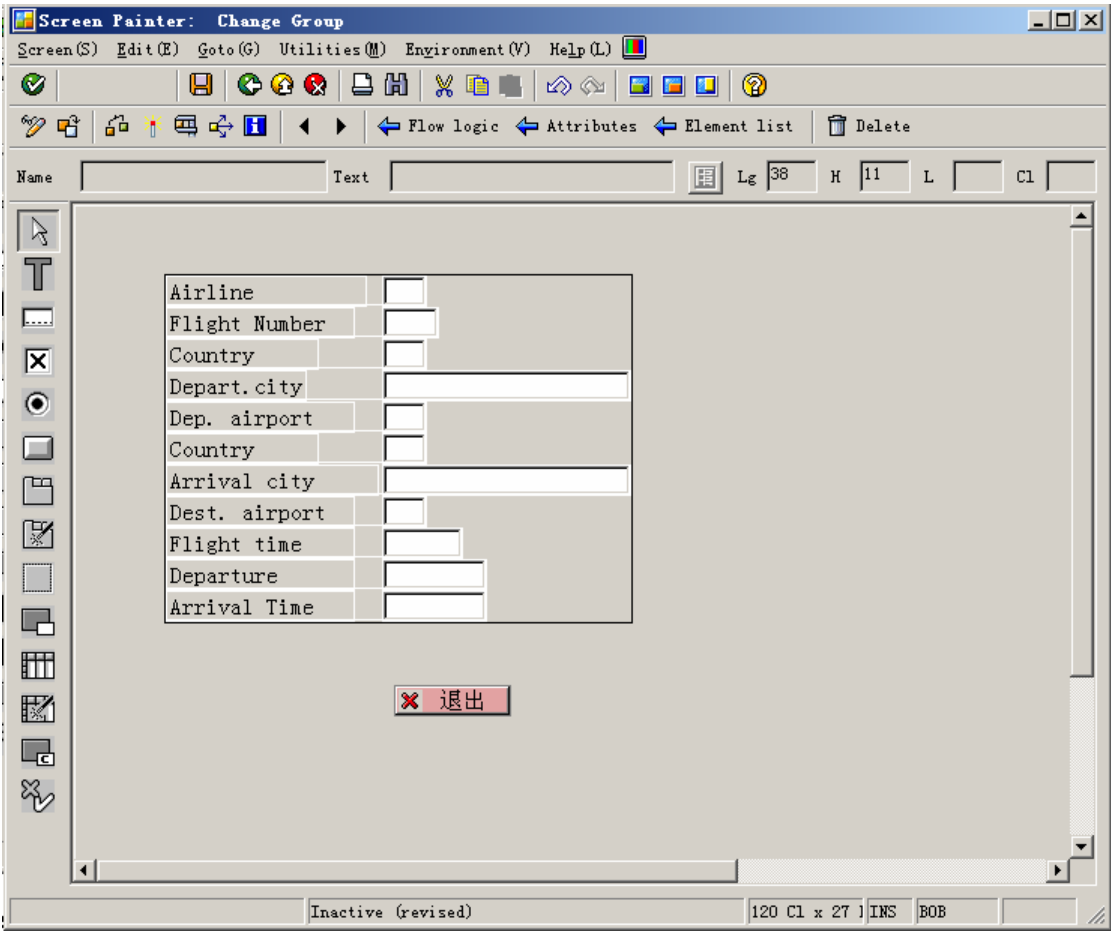


图 6-37

定义非关键字为只读，即非输入字段，如图 6-38 所示。

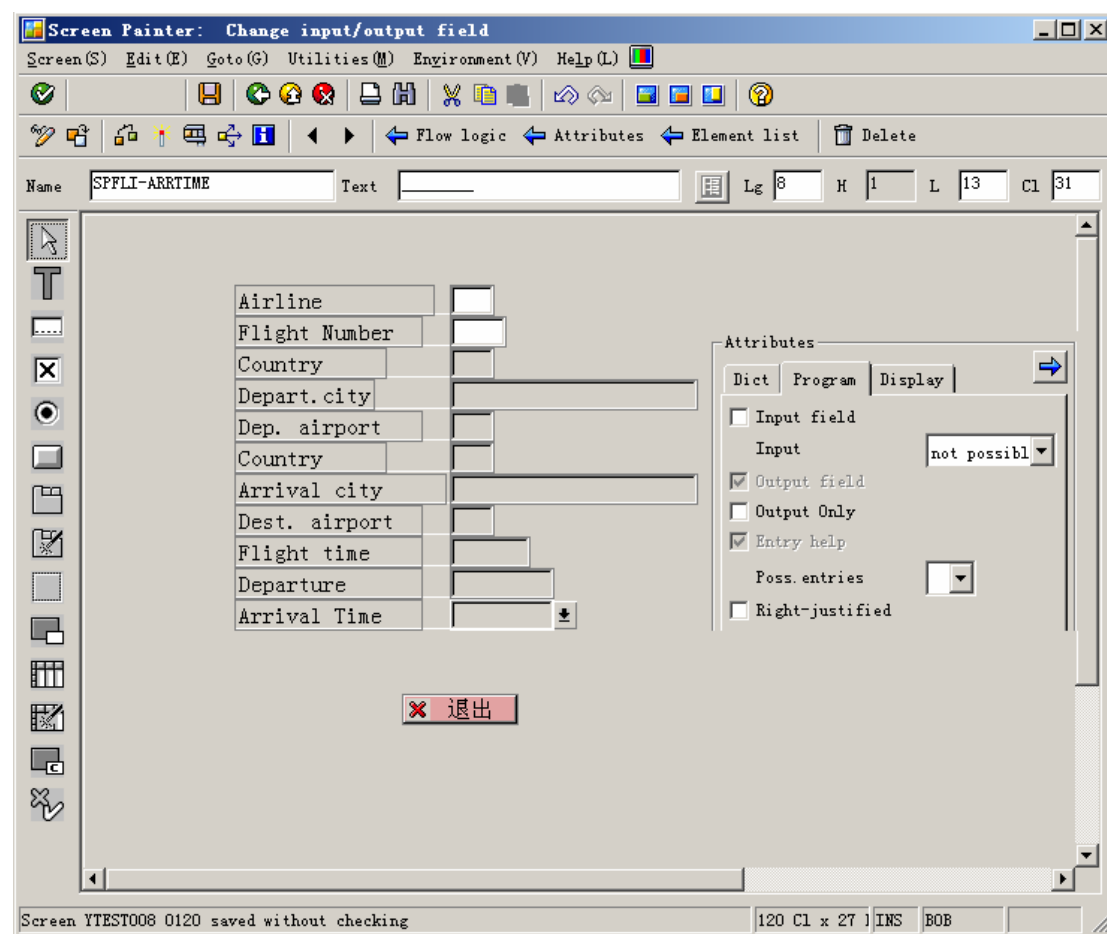


图 6-38

定义逻辑流，如图 6-39 所示。

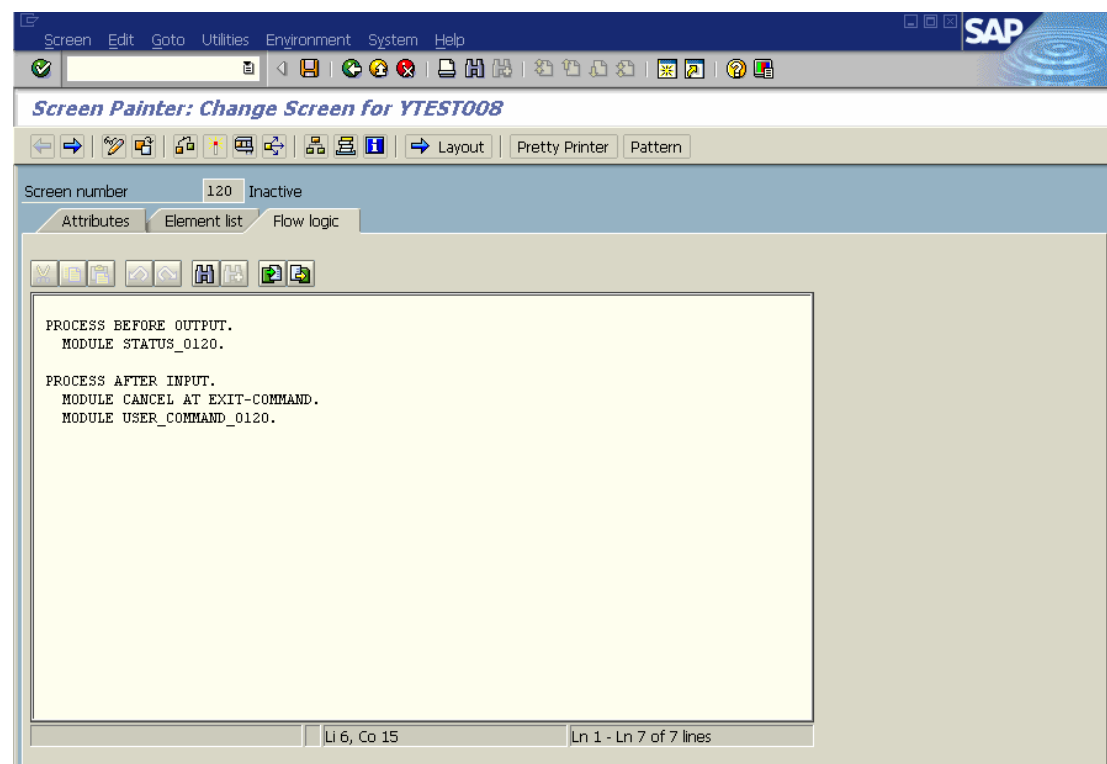


图 6-39

保存后，屏幕设计完成。

程序：

REPORT YTEST008.

*** 功能码返回值**

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

***使用航班表**

TABLES SPFLI.

***直接调用窗口**

CALL SCREEN 120.

&-----

*& Module CANCEL INPUT

&-----

*** 与屏幕逻辑流对应，退出程序**

MODULE CANCEL INPUT.

 LEAVE PROGRAM.

ENDMODULE. " CANCEL INPUT

&-----

*& Module STATUS_0120 OUTPUT

&-----

*** 定义状态条、包括菜单，工具条按钮、系统按钮等**

MODULE STATUS_0120 OUTPUT.

 SET PF-STATUS 'STATUS1'.

ENDMODULE. " STATUS_0120 OUTPUT

&-----

*& Module USER_COMMAND_0120 INPUT

&-----

*** 在输入字段或单击按钮事件后处理**

MODULE USER_COMMAND_0120 INPUT.

 SAVE_OK = OK_CODE.

 CLEAR OK_CODE.

 CASE SAVE_OK.

 WHEN 'EXIT'.

 LEAVE PROGRAM.

 ENDCASE.

***将SELECT数据赋给SPFLI记录，SPFLI记录关联屏幕字段**

 SELECT SINGLE *

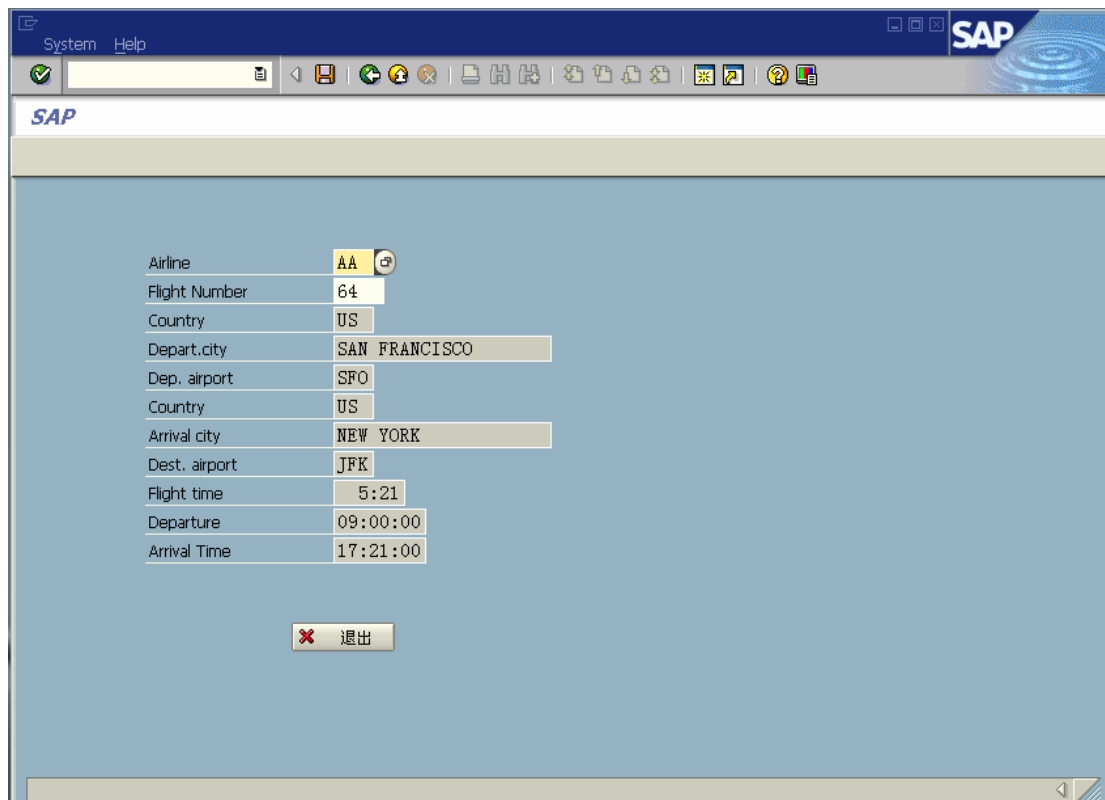
 INTO CORRESPONDING FIELDS OF SPFLI

 FROM SPFLI

```
WHERE CARRID = SPFLI-CARRID AND CONNID = SPFLI-CONNID.
IF SY-SUBRC NE 0.
  CLEAR SPFLI.
ENDIF.
ENDMODULE.                                " USER_COMMAND_0120 INPUT
```

输出：

CARRID 处输入“AA”，CONNID 处输入“64”。请先查 SPFLI 表，不要输入错误数据。回车后屏幕显示如图 6-40 所示。



The screenshot shows the SAP user command input screen. The title bar includes 'System Help' and the SAP logo. The main area displays a list of flight details for the input 'AA 64'. The details are as follows:

Airline	AA
Flight Number	64
Country	US
Depart.city	SAN FRANCISCO
Dep. airport	SFO
Country	US
Arrival city	NEW YORK
Dest. airport	JFK
Flight time	5:21
Departure	09:00:00
Arrival Time	17:21:00

At the bottom of the screen, there is a button labeled '退出' (Exit) with a red 'X' icon.

图 6-40

6.5 逻辑流

6.5.1 顺序执行逻辑流

系统是按照逻辑流的顺序执行的，例如下面的逻辑流（PAI）：
PROCESS AFTER INPUT.

```
MODULE USER_COMMAND_TEST.
```

```
  MODULE CANCEL AT EXIT-COMMAND.
```

```
  MODULE USER_COMMAND_0100.
```

如果单击退出键，那么“MODULE USER_COMMAND_0100”是不会执行的。

6.5.2 字段检查与逻辑流的控制

- 相关语法：
- 单个字段检查
FIELD<FLD1> MODULE <MDL1>.
 - 单个字段多个 MODULE 检查,如 FLD1 有两个 MODULE 检查
FIELD<FLD1> MODULE <MDL1>.
MODULE <MDL2>.
 - 检查多字段,使用 CHAIN
CHAIN.
FIELD <FLD1>.
FIELD <FLD2>,<FLD3>,<FLD4>.
MODULE <MDL 1>.
MODULE <MDL 2>.
ENDCHAIN.
表示 FLD1, FLD2, FLD3, FLD4 有 MDL1 与 MDL2 检查。
 - 不是初始值检查
FIELD <FLD1> MODLE <MDL1> ON INPUT.
ON INPUT 表示是初始值改变时执行.
有一个特殊情况:
FIELD <FLD1> MODULE <MDL1> ON *-INPUT.

表示用户输入字段首字输入‘*’, 并且输入字段属性  * Entry, MODULE 无效。
 - 有改变检查
FIELD <FLD1> MODULE <MDL1> ON REQUEST.
 - CHAIN 中有字段不是初始值检查
CHAIN.
FIELD <FLD1>.
FIELD <FLD2>,<FLD3>,<FLD4>.
MODULE <MDL 1> ON CHAIN-INPUT.
MODULE <MDL 2>.
ENDCHAIN.
.注意, CHAIN_INPUT 表示 FLD1, FLD2, FLD3, FLD4 不是初始值时执行 MDL1 检查。

6.5.3 发布消息

消息的类型见表 6-1。

表 6-1

消息类型	后继操作
E: 错误	所有字段重新输入, 重新启动 PAI 处理, 所有字段重新输入
W: 警告	所有字段允许输入, 在未输入新值回车, 也会继续 PAI 处理
I: 信息 (弹出窗口)	中断当前操作
A: 异常终止	返回

S: 成功	处理 PAI
-------	--------

建立的消息类（TCODE: SE91）YMESS 下的消息 005、006、007，如图 6-41 所示。

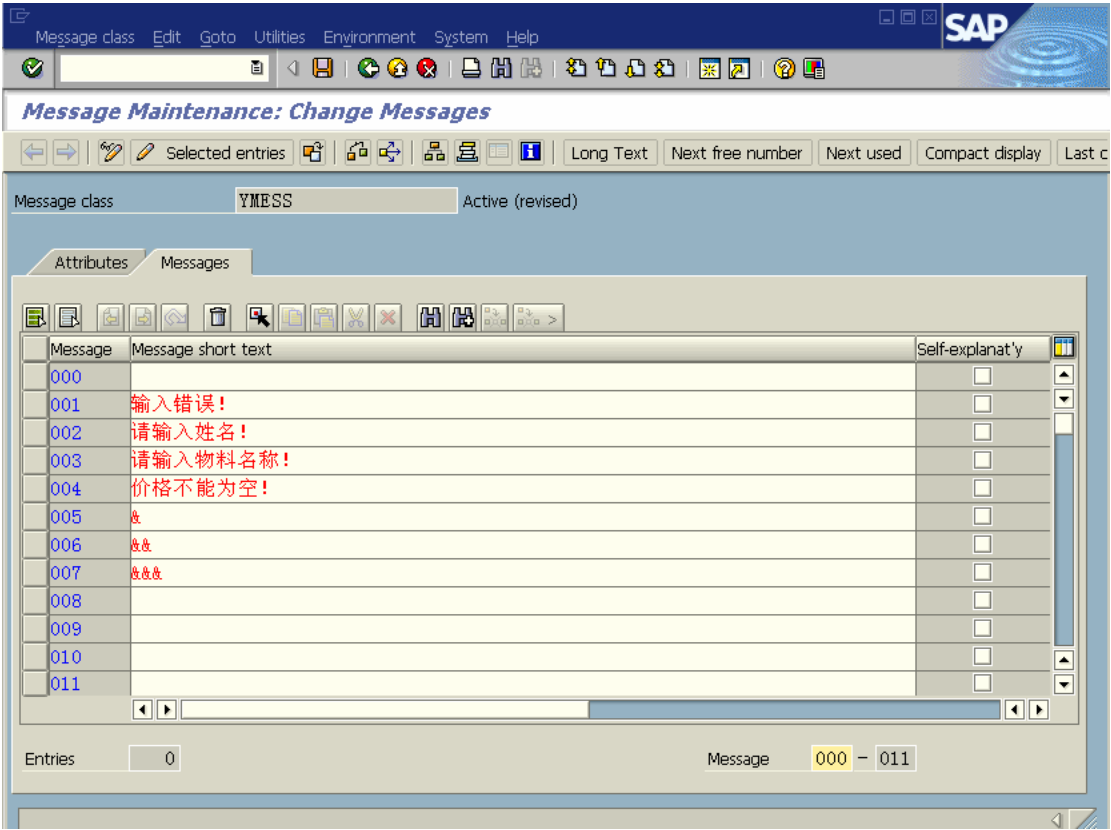


图 6-41

建立一个程序“YTEST12”，建立 Screen，加入退出按钮，并在主程序中设计退出程序。在主屏幕中加入 4 个文本字段和 4 个输入字段，如图 6-42 所示。

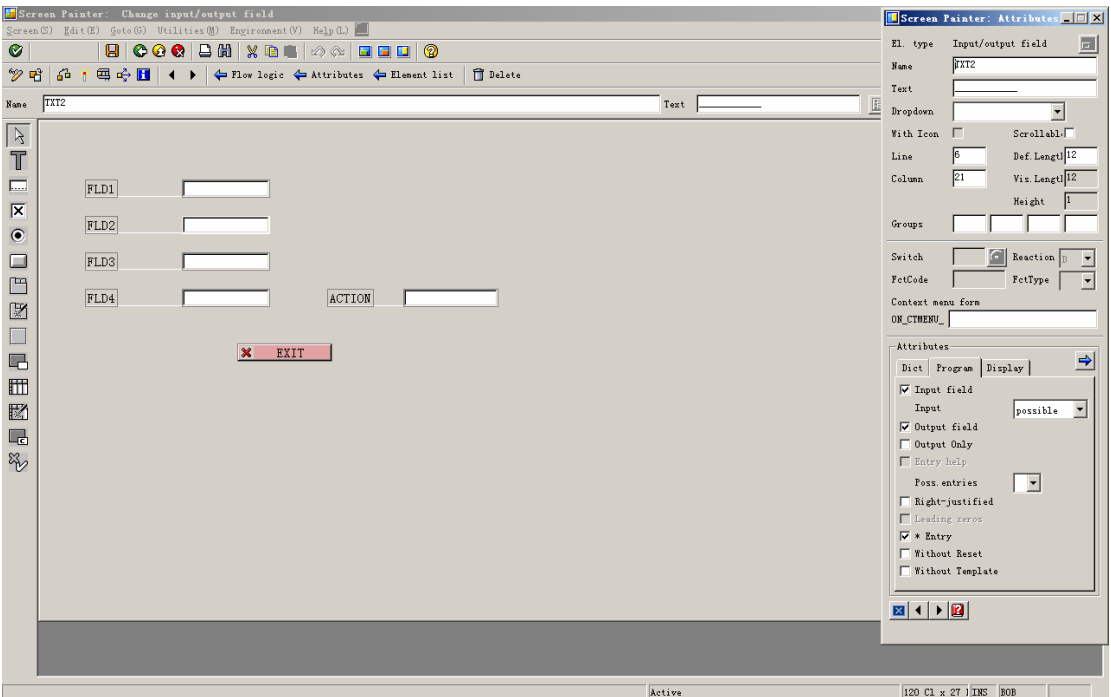



图 6-42

其中 FLD2 为  * Entry，进入逻辑流设计。

调整程序：

PROCESS BEFORE OUTPUT.

MODULE STATUS_0100.

*检查FLD1、FLD2

PROCESS AFTER INPUT.

CHAIN.

FIELD: FLD1, FLD2.

MODULE CHECK1.

ENDCHAIN.

*检查FLD3、FLD4

CHAIN.

FIELD: FLD3, FLD4.

MODULE CHECK2.

ENDCHAIN.

*当FLD2的首字符是*时执行CHECK3检查

FIELD FLD2 MODULE CHECK3 ON *-INPUT.

MODULE CANCEL AT EXIT-COMMAND.

MODULE USER_COMMAND_0100.

调整主程序：

REPORT YTEST12.

*功能码返回值

DATA: OK_CODE TYPE SY-UCOMM,

SAVE_OK TYPE SY-UCOMM.

*定义输入字段变量

DATA: FLD1(20) TYPE C,

FLD2(20) TYPE C,

FLD3(20) TYPE C,

FLD4(20) TYPE C,

ACTION(50) TYPE C.

*直接调用窗口

CALL SCREEN 100.

&-----

*& Module CANCEL INPUT

&-----

* CANCEL MODULE 与屏幕逻辑流对应，退出程序

MODULE CANCEL INPUT.

LEAVE PROGRAM.

ENDMODULE. " CANCEL INPUT

&-----

*& Module STATUS_0100 OUTPUT

&-----

* 定义状态条, 包括菜单、工具条按钮、系统按钮等

```
MODULE STATUS_0100 OUTPUT.
    SET PF-STATUS 'STATUS1'.
```

```
ENDMODULE.                " STATUS_0100  OUTPUT
```

&-----

```
*&      Module  USER_COMMAND_0100  INPUT
```

&-----

* 如果能执行PAI, 把FLD1、FLD2、FLD3、FLD4合并到ACTION输入字段

```
MODULE USER_COMMAND_0100 INPUT.
```

```
    SAVE_OK = OK_CODE.
```

```
    CLEAR OK_CODE.
```

```
    CASE SAVE_OK.
```

```
        WHEN 'EXIT'.
```

```
            LEAVE PROGRAM.
```

```
    ENDCASE.
```

```
    ACTION = ''.
```

```
    CONCATENATE FLD1 FLD2 FLD3 FLD4 INTO ACTION.
```

```
ENDMODULE.                " USER_COMMAND_0100  INPUT
```

&-----

```
*&      Module  CHECK1  INPUT
```

&-----

* 输入FLD1、FLD2 时检查

```
MODULE CHECK1 INPUT.
```

*当输入FLD1 = 'FLD1' 时提示错误, 而且不执行PAI

```
    IF FLD1 = 'FLD1'.
```

```
        MESSAGE E005(YMESS) WITH 'FLD1 ERROR'.
```

```
    ENDIF.
```

*当输入FLD1 = 'FLD11' 时弹出窗口提示, 执行PAI

```
    IF FLD1 = 'FLD11'.
```

```
        MESSAGE I005(YMESS) WITH 'FLD1 WARNING'.
```

```
    ENDIF.
```

*当输入FLD1 = 'FLD111' 时输出成功信息, 执行PAI

```
    IF FLD1 = 'FLD111'.
```

```
        MESSAGE S005(YMESS) WITH 'FLD1 OK'.
```

```
    ENDIF.
```

```
ENDMODULE.                " CHECK1  INPUT
```

&-----

```
*&      Module  CHECK2  INPUT
```

&-----

* 输入FLD3、FLD4 时检查

```

*-----*
MODULE CHECK2 INPUT.
  IF FLD3 = 'FLD3'.
    MESSAGE E005(YMESS) WITH 'FLD3 ERROR'.
  ENDIF.
  IF FLD3 = 'FLD33'.
    MESSAGE I005(YMESS) WITH 'FLD3 WARNING'.
  ENDIF.
  IF FLD3 = 'FLD333'.
    MESSAGE S005(YMESS) WITH 'FLD3 OK'.
  ENDIF.
ENDMODULE.                " CHECK2  INPUT
*&-----*
*&      Module  CHECK3  INPUT
*&-----*
*当FLD2第一个字符是'*'时,执行该MODULE,弹出信息提示
*-----*
MODULE CHECK3 INPUT.
  MESSAGE I005(YMESS) WITH 'FLD2 * WARNING'.
ENDMODULE.                " CHECK3  INPUT

```

输出如图 6-43 所示。

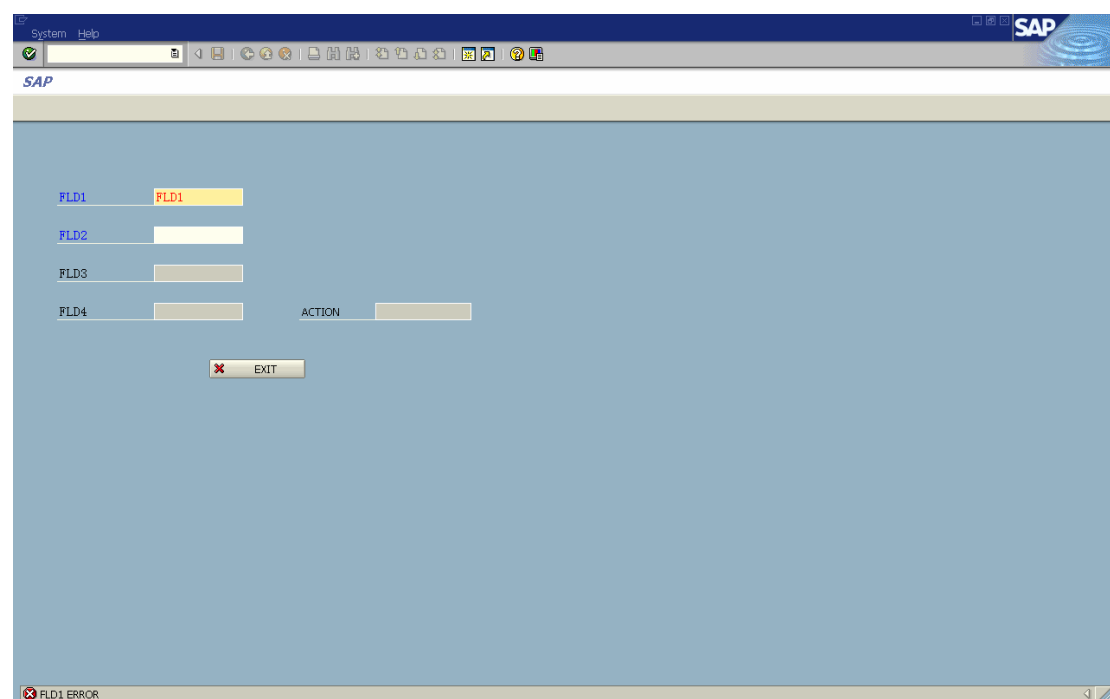


图 6-43

当输入 FLD1 = 'FLD1' 时，提示错误，并且变量 ACTION 不会更改，
当输入 FLD1 = 'FLD11' 时，弹出提示框，确认后变量 ACTION 更改，如图 6-44 所示。

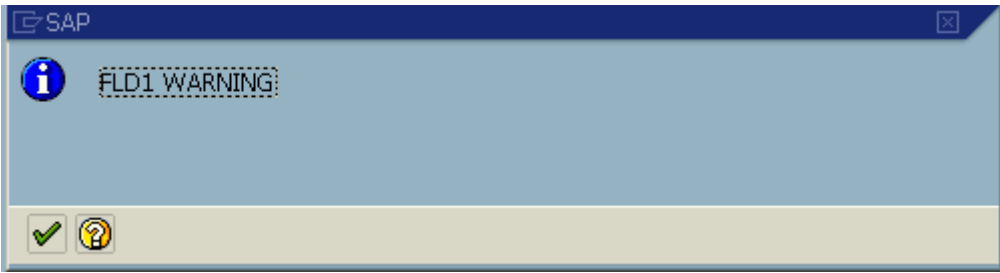


图 6-44

当 FLD2 首字符为 “*” 时，出现提示框，如图 6-45 所示。

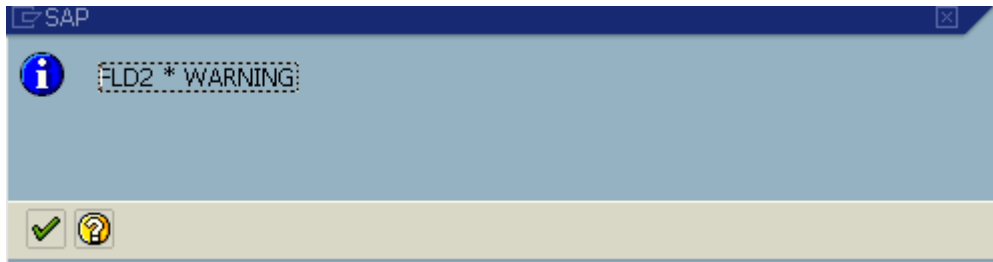


图 6-45

变量 ACTION 会更新。

6.6 Listbox 下拉框设计

在屏幕上建立一个输入输出 FIELD,设定属性下拉：Listbox，如图 6-46 所示。

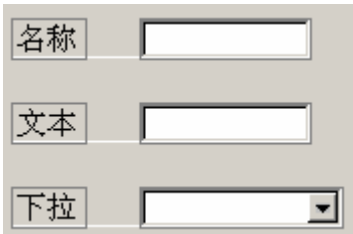


图 6-46

程序:

```
REPORT  YTEST13.
*功能码返回值
DATA: OK_CODE TYPE SY-UCOMM,
      SAVE_OK TYPE SY-UCOMM.
*VRM类型定义接口
TYPE-POOLS VRM.
DATA: FLD1(20) TYPE C.
*内表、记录，字段变量定义
DATA: FNAME TYPE VRM_ID,
      VVA    TYPE VRM_VALUES,
      LVVA   LIKE LINE OF VVA.
*直接调用窗口
CALL SCREEN 100.
*&-----*
```

```

*&      Module  STATUS_0100  OUTPUT
*&-----*
* 定义状态条, 包括菜单、工具条按钮、系统按钮等
* 直接在PBO中将值填入内表, 再设定下拉框
*-----*
MODULE STATUS_0100 OUTPUT.
  SET PF-STATUS 'STATUS1'.
*加两条记录
  FNAME = 'FLD1'.
  LVVA-KEY = '1'.
  LVVA-KEY = '广州'.
  APPEND LVVA TO VVA.
  CLEAR LVVA.
  FNAME = 'FLD1'.
  LVVA-KEY = '上海'.
  APPEND LVVA TO VVA.
  CALL FUNCTION 'VRM_SET_VALUES'
    EXPORTING
      ID                = FNAME
      VALUES           = VVA
*  EXCEPTIONS
*    ID_ILLEGAL_NAME    = 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.

ENDMODULE.                                " STATUS_0100  OUTPUT
*&-----*
*&      Module  CANCEL  INPUT
*&-----*
* CANCEL MODULE 与屏幕逻辑流对应, 退出程序
*-----*
MODULE CANCEL INPUT.
  LEAVE PROGRAM.
ENDMODULE.                                " CANCEL  INPUT
*&-----*
*&      Module  USER_COMMAND_0100  INPUT
*&-----*
* PAI
*-----*
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

输出结果如图 6-47 所示。



图 6-47

6.7 修改屏幕状态

可以在程序运行时动态地设置对象的属性，如设置一个字段是否有效，在 Screen 设计时，可以将字段分组，在 PBO 事件中设置属性。

本节在屏幕上建立一个 CHECKBOX、若干输入字段，将字段分成两个组，每一个组有一个 BUTTON，单击组 BUTTON，根据 CHECKBOX 的值更改字段组的属性，设计界面如图 6-48 所示。



图 6-48

1-1 和 1-2 为 NO1 组，2-1 和 2-2 为 NO2 组。

程序:

```
REPORT YTEST14.  
*功能码返回值  
DATA: OK_CODE TYPE SY-UCOMM,  
      SAVE_OK TYPE SY-UCOMM.  
  
DATA: SGRP(20) TYPE C,  
      SMOD(1) TYPE C.  
*是否修改
```

```

DATA: CANMOD(1) TYPE C.
*默认可以修改
CANMOD = 'X'.
*直接调用窗口
CALL SCREEN 100.
*&-----*
*&      Module  STATUS_0100  OUTPUT
*&-----*
* 定义状态条、包括菜单、工具条按钮、系统按钮
*-----*
MODULE STATUS_0100 OUTPUT.
  SET PF-STATUS 'STATUS1'.
*根据按钮确定字段组
  SGRP = ''.
  IF SAVE_OK = 'BTN1'.
    SGRP = 'N01'.
  ELSE.
    SGRP = 'N02'.
  ENDIF.
*CHECKBOX转换
  IF CANMOD = 'X'.
    SMOD = '1'.
  ELSE.
    SMOD = '0'.
  ENDIF.
*逐个扫描屏幕元素
  LOOP AT SCREEN.
*如果是需要修改的组, 更改变量, 其他组不考虑, 如果正式考虑先存储到变量
    IF SCREEN-GROUP1 = SGRP.
      SCREEN-INPUT = SMOD.
*更新到窗口元素变量
      MODIFY SCREEN.
    ENDIF.
  ENDLOOP.
ENDMODULE.                                " STATUS_0100  OUTPUT
*&-----*
*&      Module  CANCEL  INPUT
*&-----*
* CANCEL MOUDLE 与屏幕逻辑流对应, 退出程序
*-----*
MODULE CANCEL INPUT.
  LEAVE PROGRAM.
ENDMODULE.                                " CANCEL  INPUT
*&-----*

```

```

*&      Module  USER_COMMAND_0100  INPUT
*&-----*
*  PAI
*-----*
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
    
```


输出结果如图 6-49 所示。

图 6-49

当选择不更改，单击设定组 1 按钮时，测试组 1 状态变为不能编辑。

6.6 子窗口

一个程序可以有多个窗口，通过子窗口，可以把若干个子窗口放在一个窗口上，这样，既保持了整体性，又方便对单个屏幕的维护。在 SAP 业务程序中，大量地使用了子窗口设计。

窗口对象子屏幕范围（) 对象用来定义一个子窗口在主窗口上的大小、位置等属性，通过逻辑流和主程序将对象关联到子窗口。

例如：建立一个主窗口和两个子窗口，在主窗口上再创建一个子窗口范围，增加两个按钮，单击按钮时切换子窗口，设计界面如图 6-50 所示。

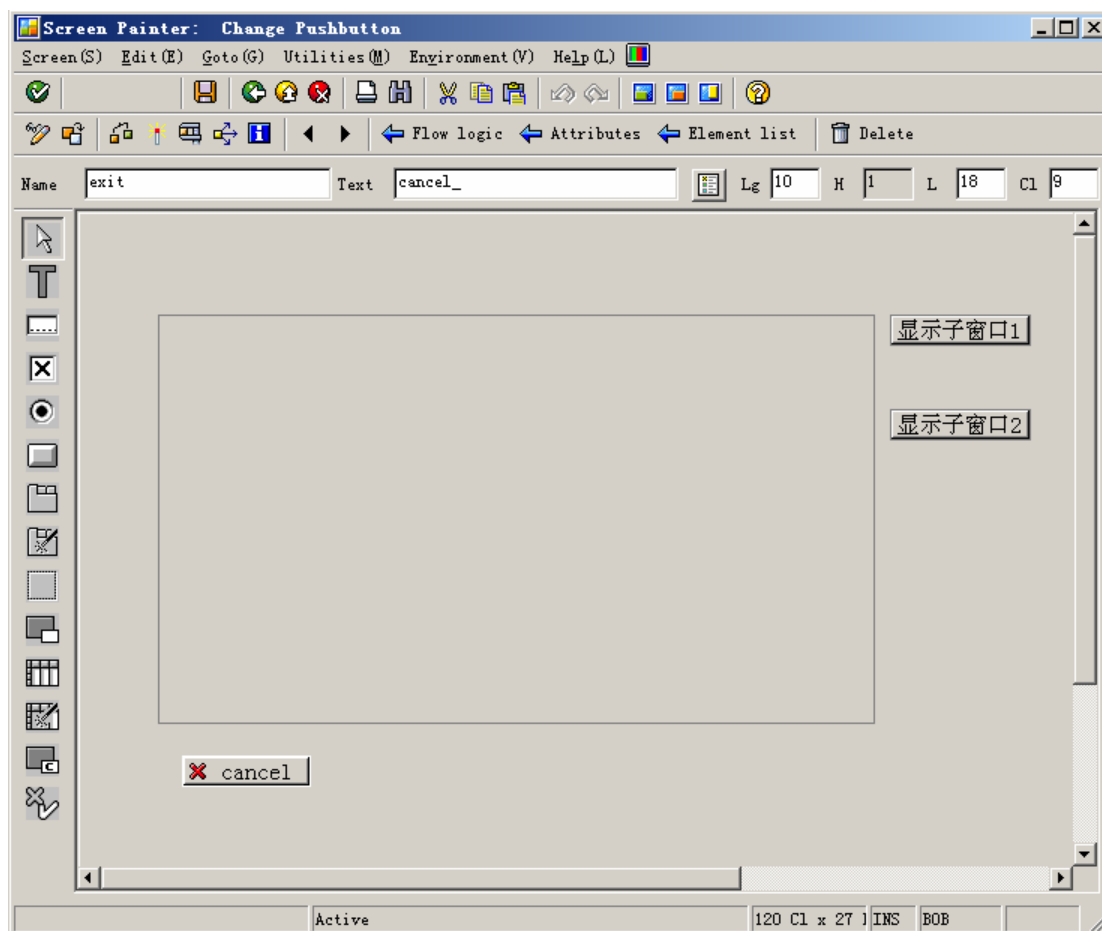


图 6-50

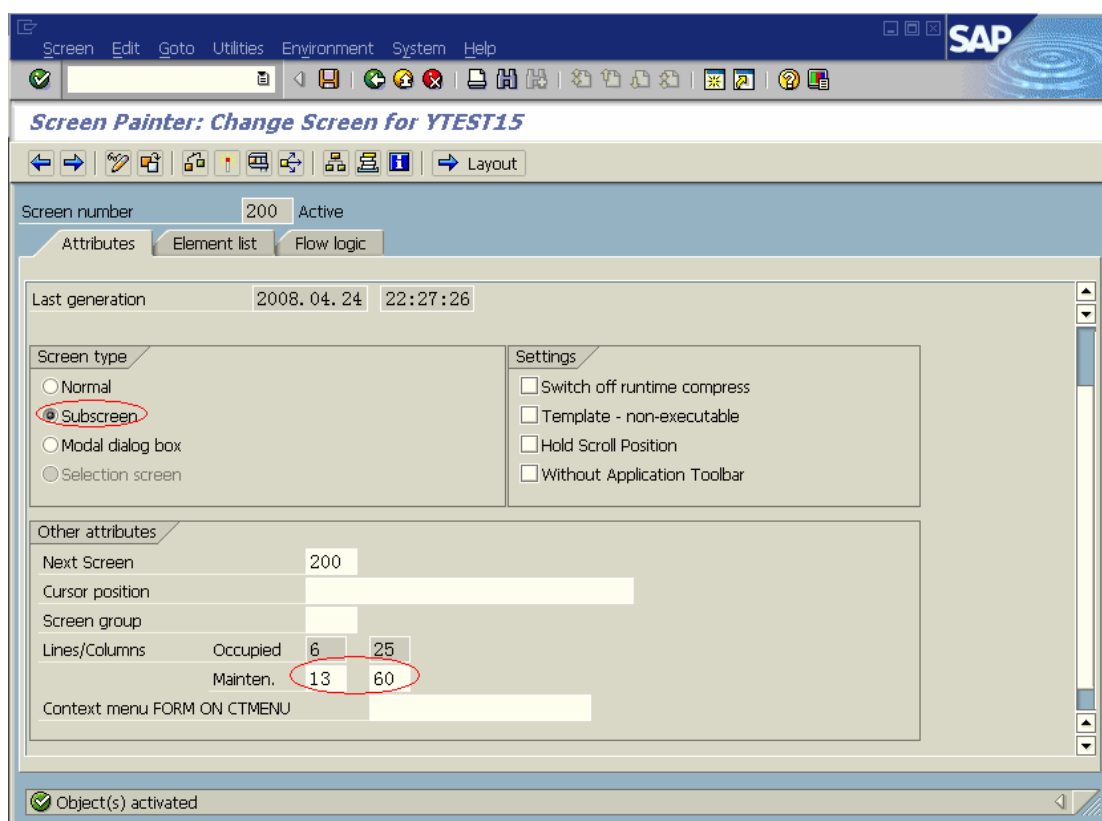


图 6-51

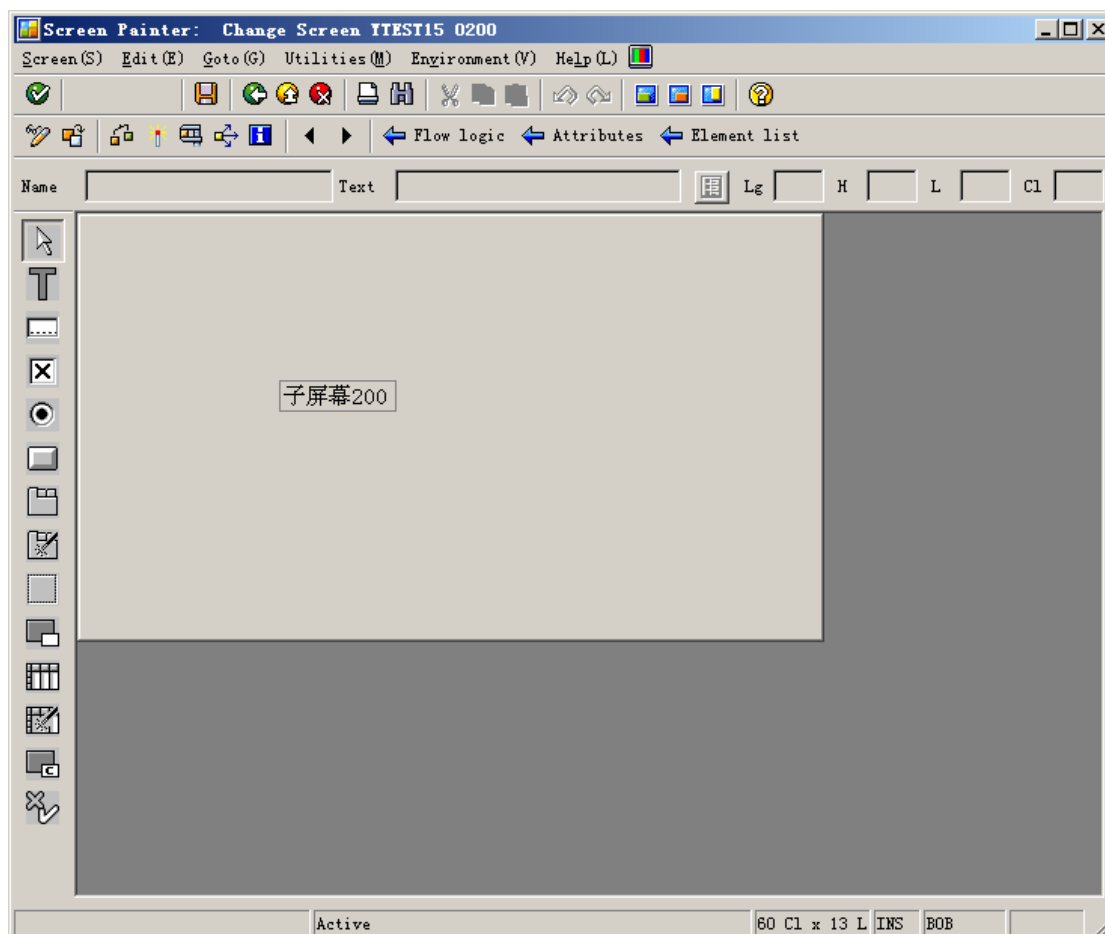


图 6-52

定义好子屏幕 300。

回到主屏幕，定义逻辑流：

PROCESS BEFORE OUTPUT.

MODULE STATUS_0100.

*SUBSCR是定义的子屏幕范围的名称

CALL SUBSCREEN: SUBSCR INCLUDING SY-REPID SID1.

PROCESS AFTER INPUT.

MODULE CANCEL AT EXIT-COMMAND.

MODULE USER_COMMAND_0100.

*使子屏幕显示

CALL SUBSCREEN SUBSCR.

主程序：

REPORT YTEST15 .

*功能码返回值

DATA: OK_CODE TYPE SY-UCOMM,

SAVE_OK TYPE SY-UCOMM.

*子窗口默认用200子窗口

DATA: SID1(4) TYPE N VALUE '200'.

*直接调用窗口

CALL SCREEN 100.

```

*&-----*
*&      Module  STATUS_0100  OUTPUT
*&-----*
*      text
*-----*
MODULE STATUS_0100 OUTPUT.
  SET PF-STATUS 'STATUS1'.

ENDMODULE.                " STATUS_0100  OUTPUT
*&-----*
*&      Module  CANCEL  INPUT
*&-----*
*      text
*-----*
MODULE CANCEL INPUT.
  LEAVE PROGRAM.
ENDMODULE.                " CANCEL  INPUT
*&-----*
*&      Module  USER_COMMAND_0100  INPUT
*&-----*
*      text
*-----*
MODULE USER_COMMAND_0100 INPUT.
  SAVE_OK = OK_CODE.
  CLEAR OK_CODE.
  CASE SAVE_OK.
    *当单击第一个按钮，显示200子窗口
    WHEN 'BTN1'.
      SID1 = '200'.
    *当单击第二个按钮，显示300子窗口
    WHEN 'BTN2'.
      SID1 = '300'.
    WHEN 'EXIT'.
      LEAVE PROGRAM.
  ENDCASE.
ENDMODULE.                " USER_COMMAND_0100  INPUT

```

输出结果如图6-53所示。

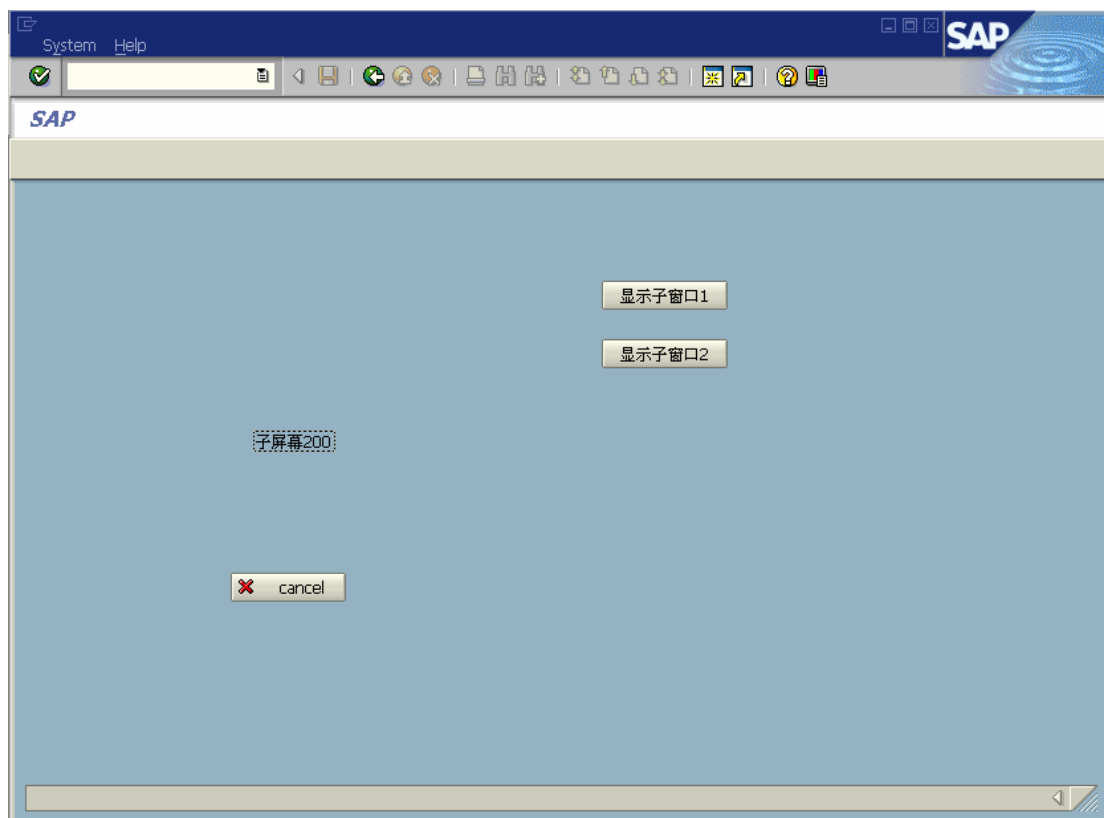




图 6-53

当单击“显示子窗口 2”按钮时，显示子窗口 300，如图 6-54 所示。




图 6-54

6.9 表条目控制

表条目控制是实现将多个子屏幕以页面形式切换的控件，有手工制作和向导生成两种，在本节中，不再重复子屏幕创建操作，只表达操作上的特殊性，手工制作的图标是“”向导制作的图标是“”。

6.9.1 手工制作

进入屏幕设计，单击按钮，创建一个表条目控制，如图 6-55 所示。

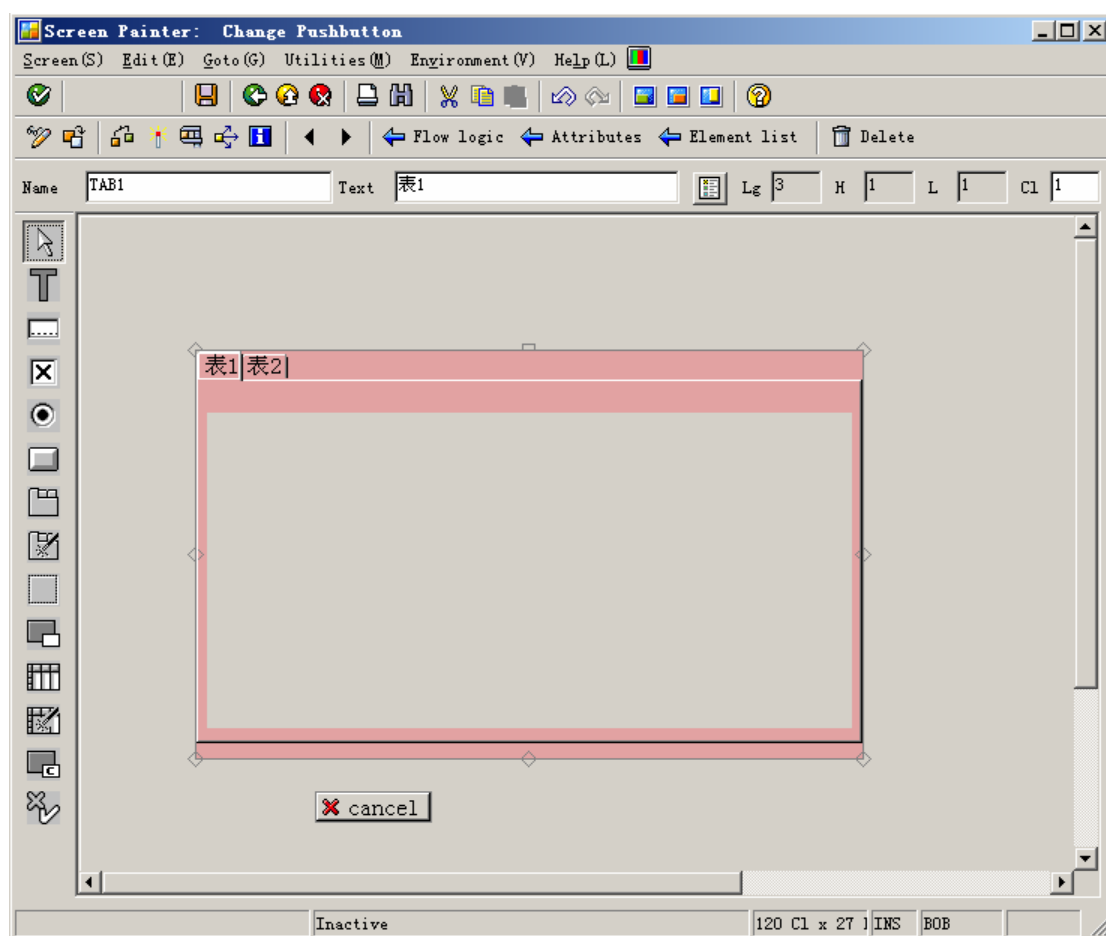


图 6-55

图 6-55 中控件显示粉红色表示其属性未设置完成。

逐页设置名称、文本、功能码属性，接着选中“子屏幕范围”按钮，在第一页创建该对象，设定好子屏幕的属性后，第一页就设置完成，不再显示粉红色，如图 6-56 所示的“表 1”。

6.9.2 向导制作

在设计屏幕上创建“表条目控制制作向导”对象，启动向导程序，如图 6-58 所示。

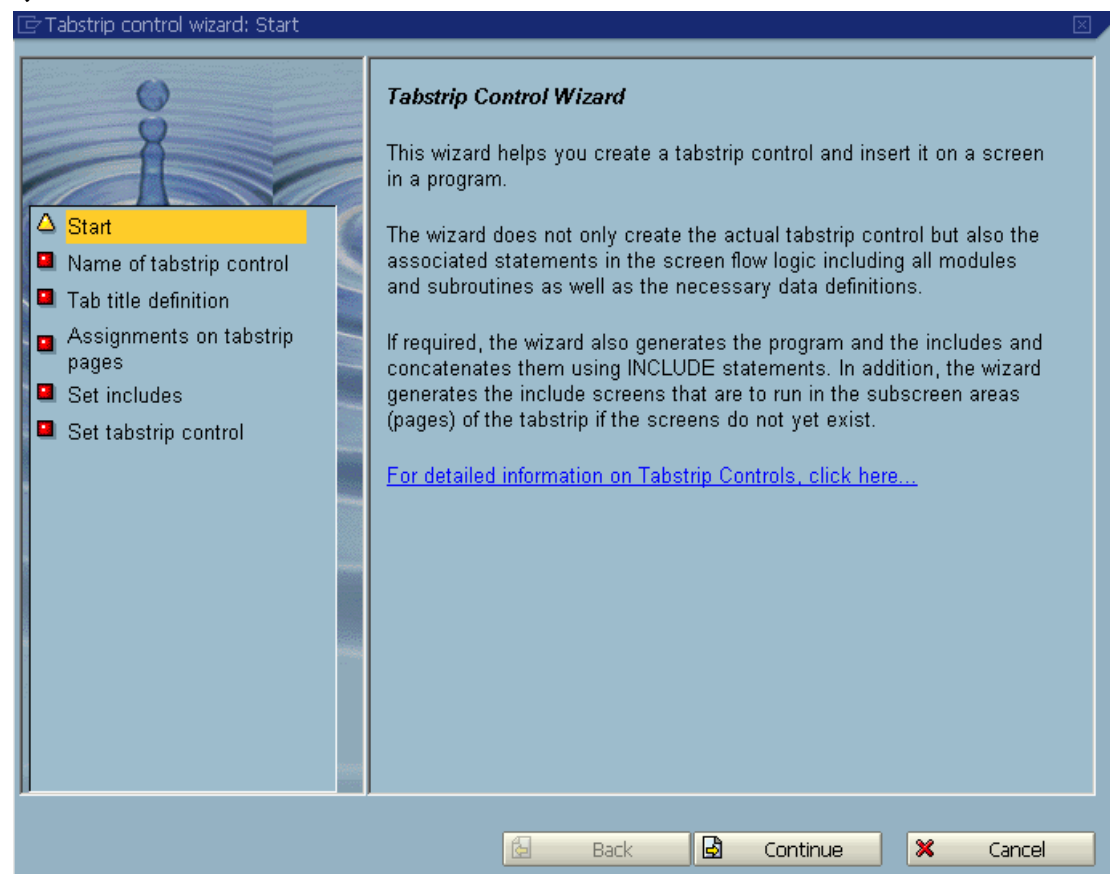


图 6-58

输入名称后继续，如图 6-59 所示。

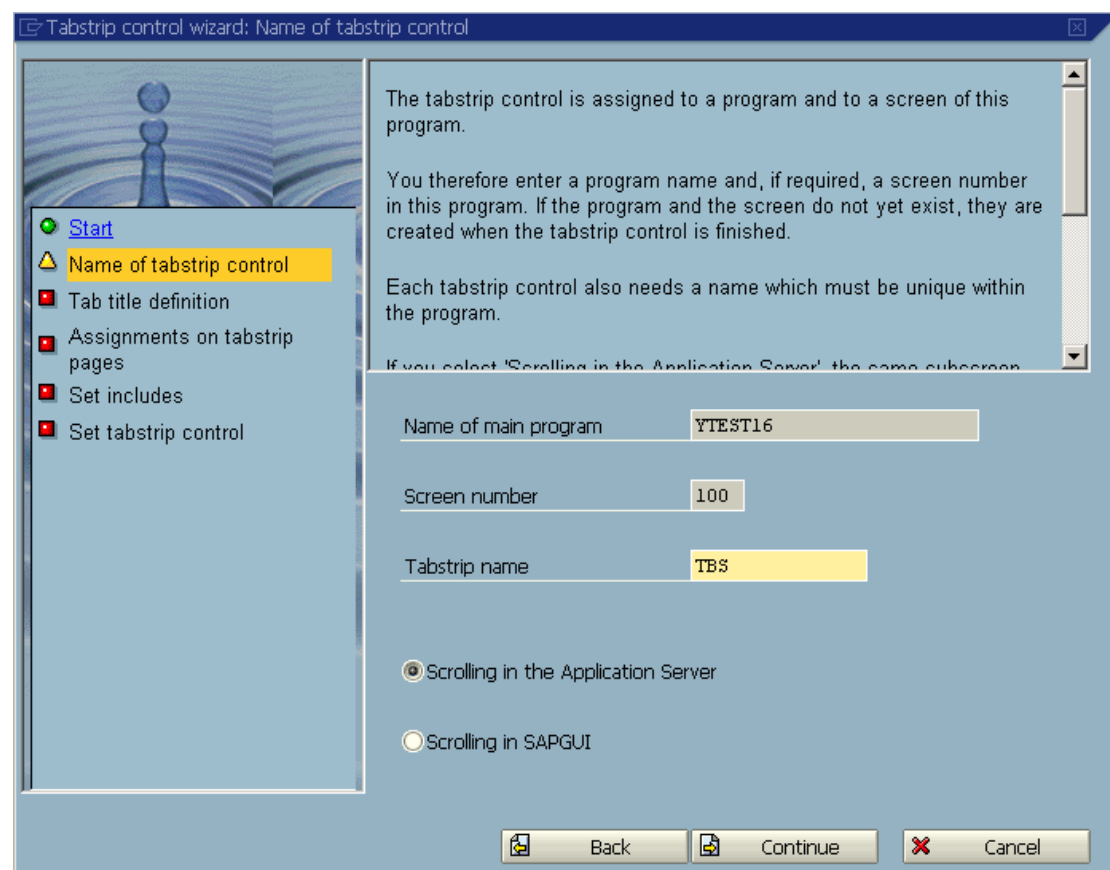


图 6-59

输入需要建立页面的名称，继续，如图 6-60 所示。

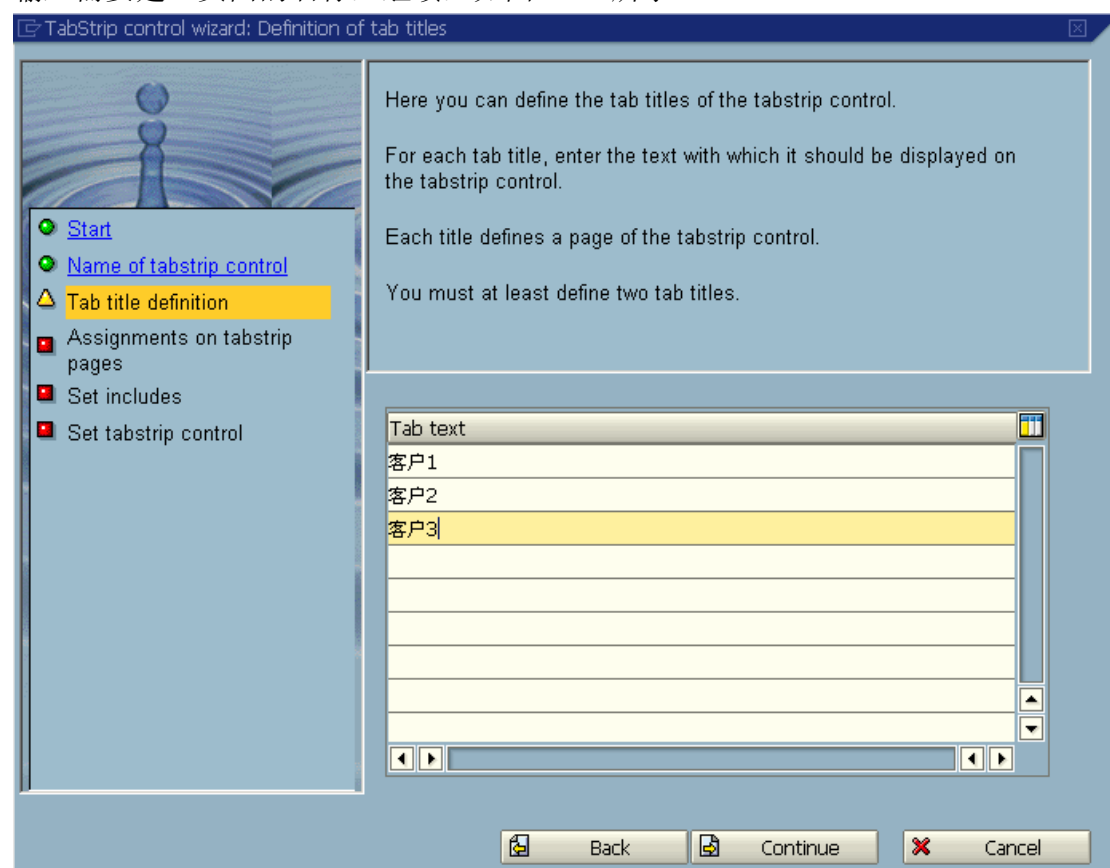
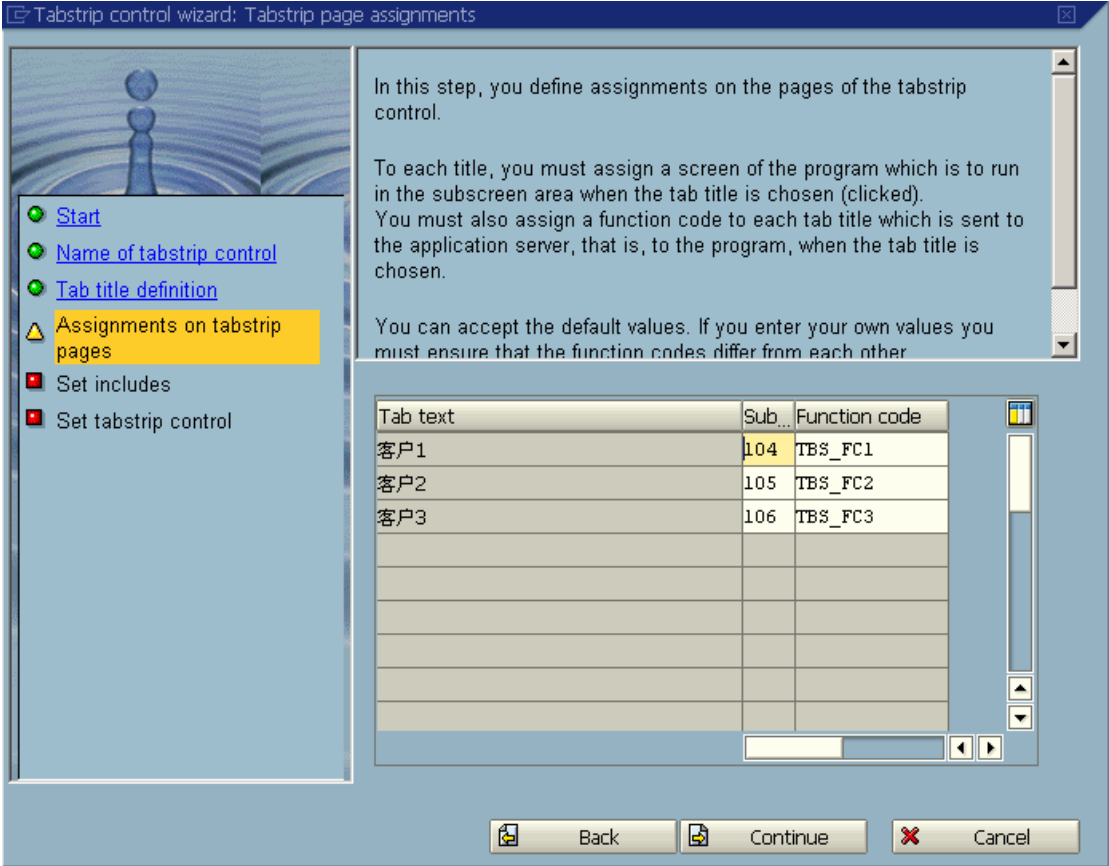


图 6-60

向导产生页面子屏幕编号和每页功能码，可以在此基础上调整，继续，如图 6-61 所示。



默认调用主屏幕的主程序名称，继续，如图 6-62 所示。

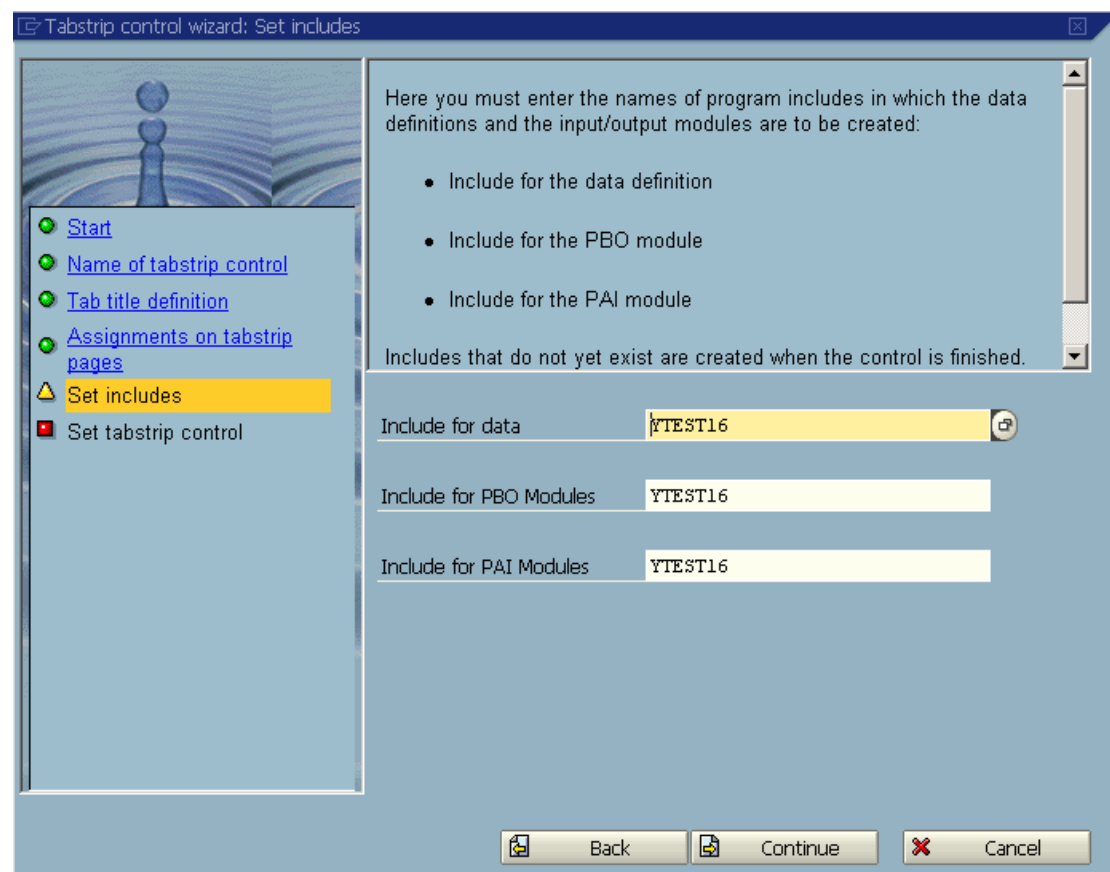


图 6-62

系统提示所示设定参数信息，单击“完全”按钮，如图 6-63 所示。

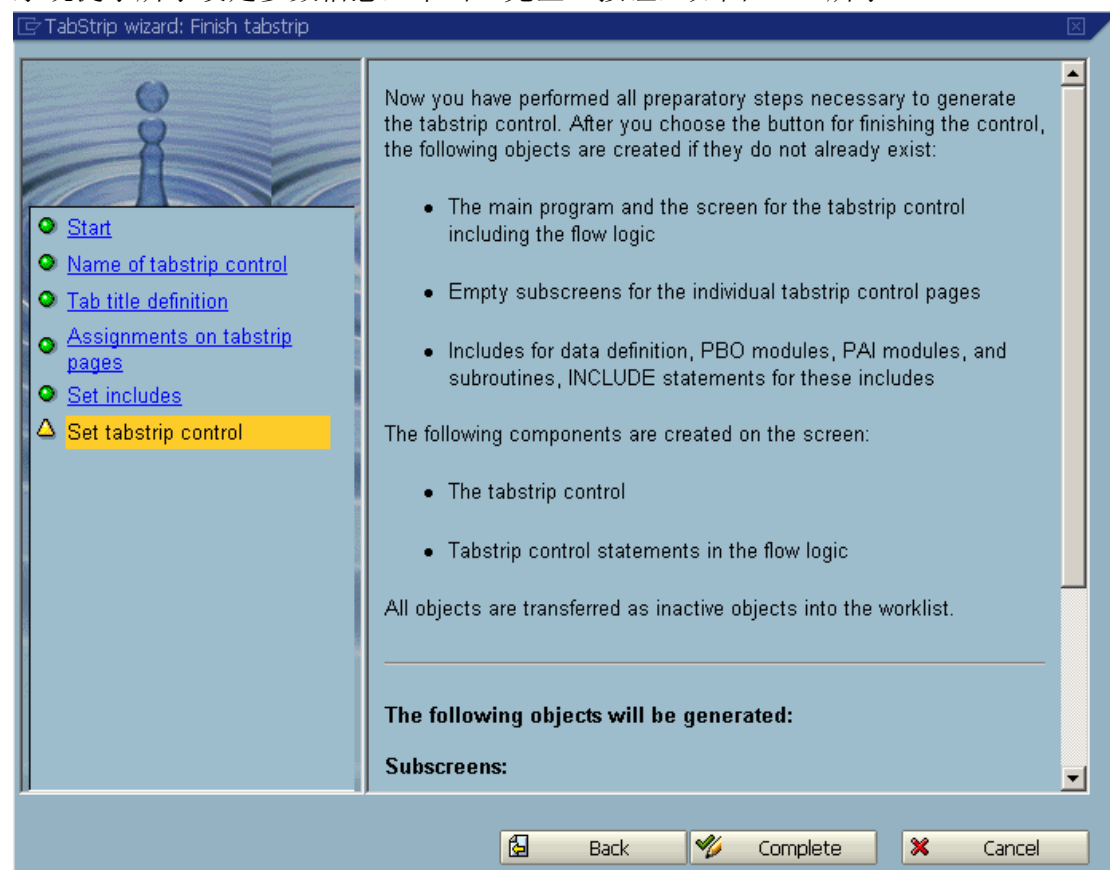


图 6-63

打开设计界面，系统自动建立完成全部页面、逻辑流和主程序代码，如图 6-64 所示。

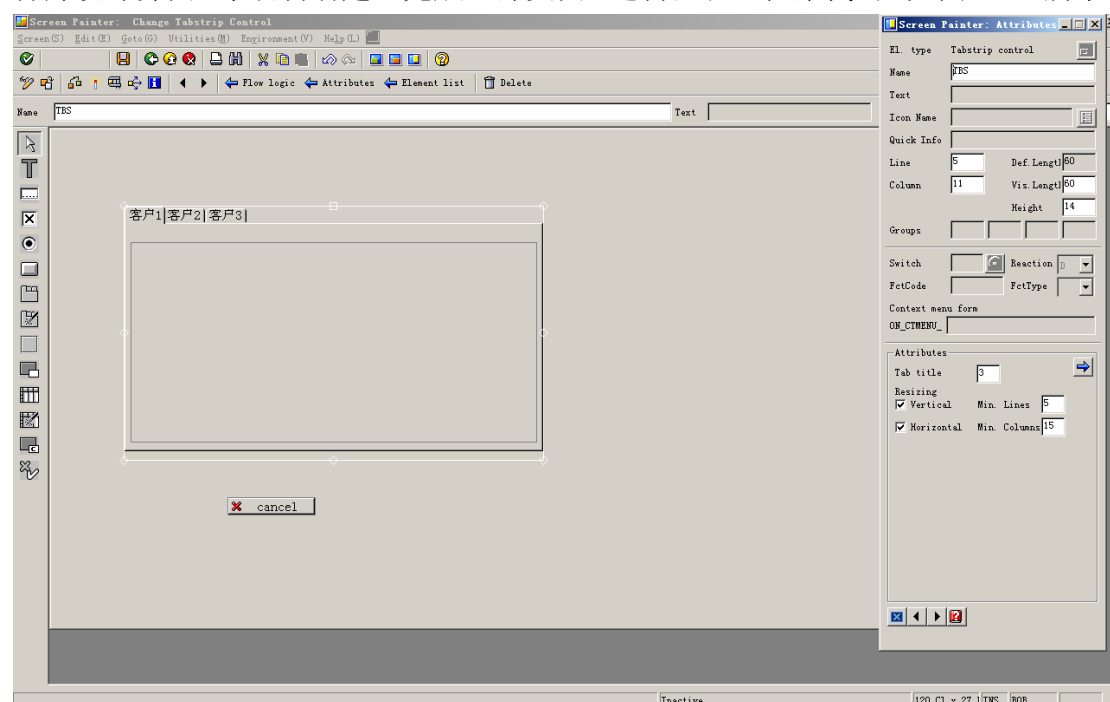



图 6-64

6.10 通过定制控制在屏幕上显示图片

可以通过“定制”对象在屏幕上显示各种控件，如图片、Tree Control、ALV、编辑器等，后面的实例将会用到这个控件。先介绍在一个定制控制对象内显示图片。

6.10.1 图片的上载

通过 TCODE: SMW0 上传图片，选择应用程序的二进制数据，单击查找  按钮，如图 6-65 所示。

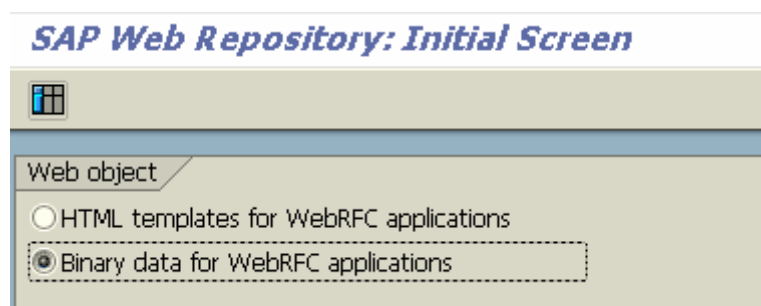


图 6-65

开发类选择“SLIS”（系统中已存在的开发类），如图 6-66 所示。

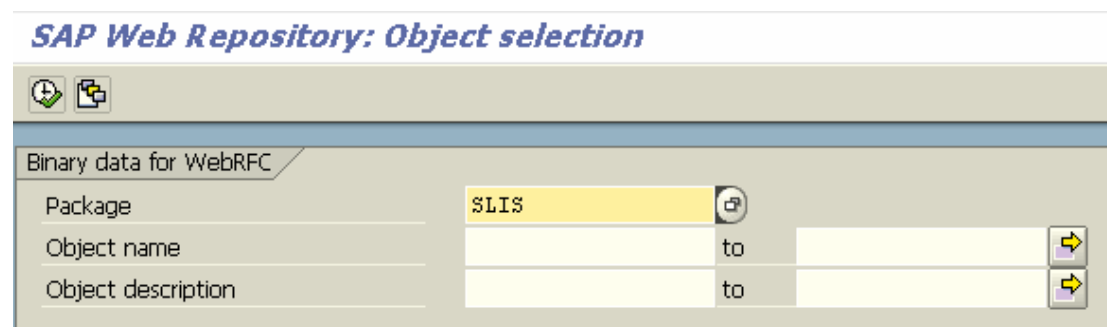



图 6-66

屏幕显示已有对象，单击新增按钮（），如图 6-67 所示。

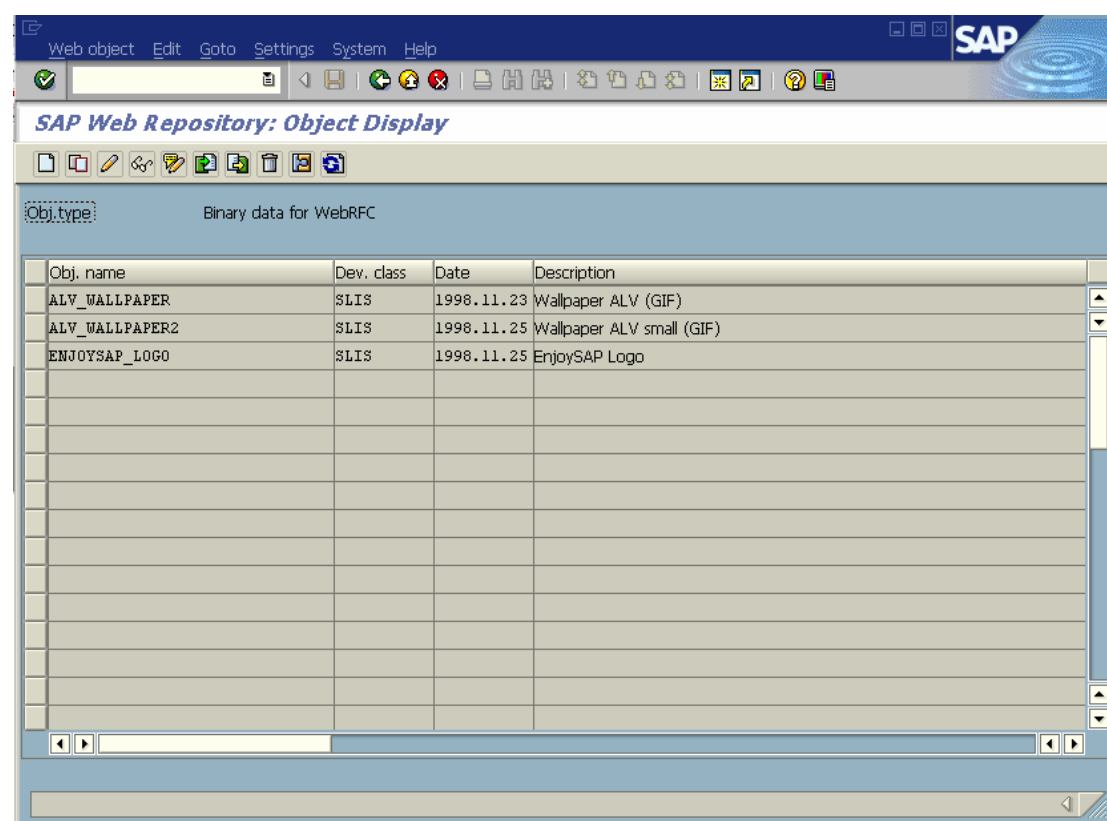



图 6-67

输入对象名称，单击输入按钮（），如图 6-68 所示。

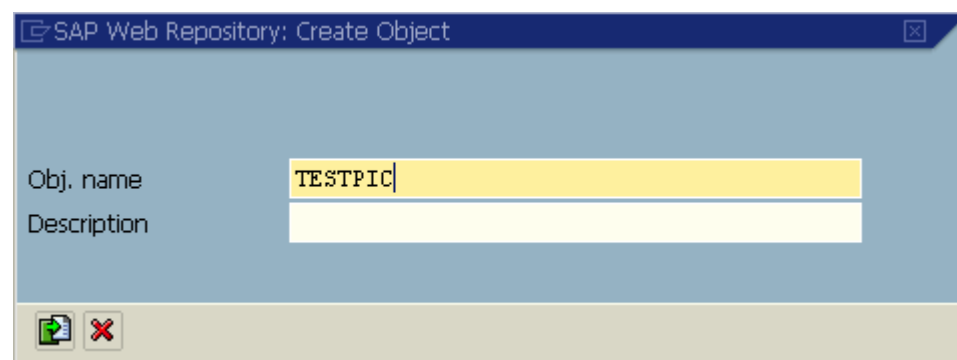


图 6-68

选择开发类后，单击保存按钮，如图 6-70 所示。

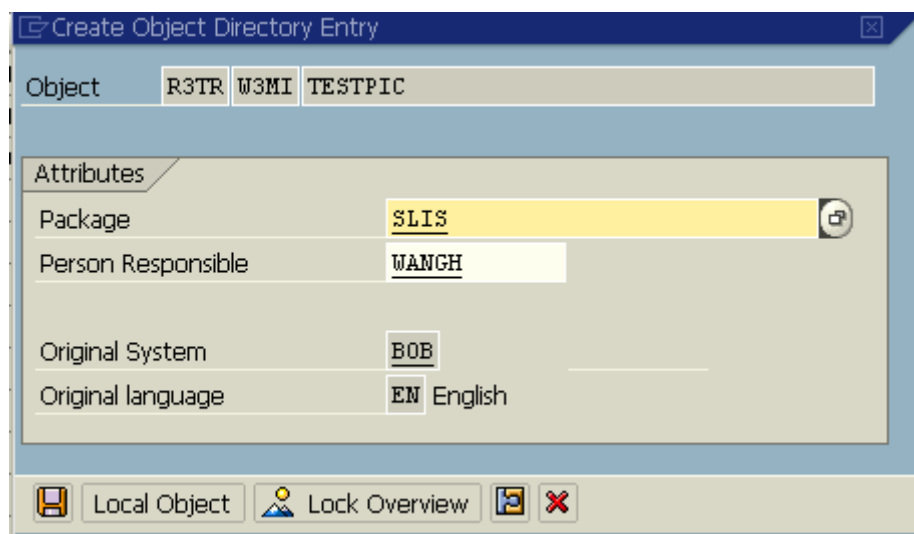


图 6-70

产生传输入请求，单击按钮 (✓)，如图 6-71 所示。

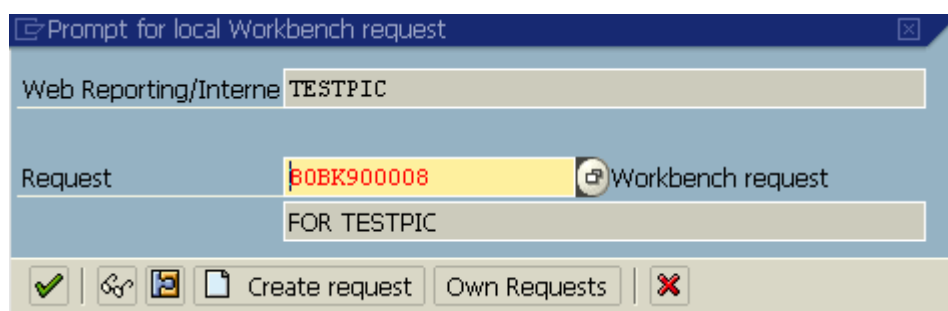
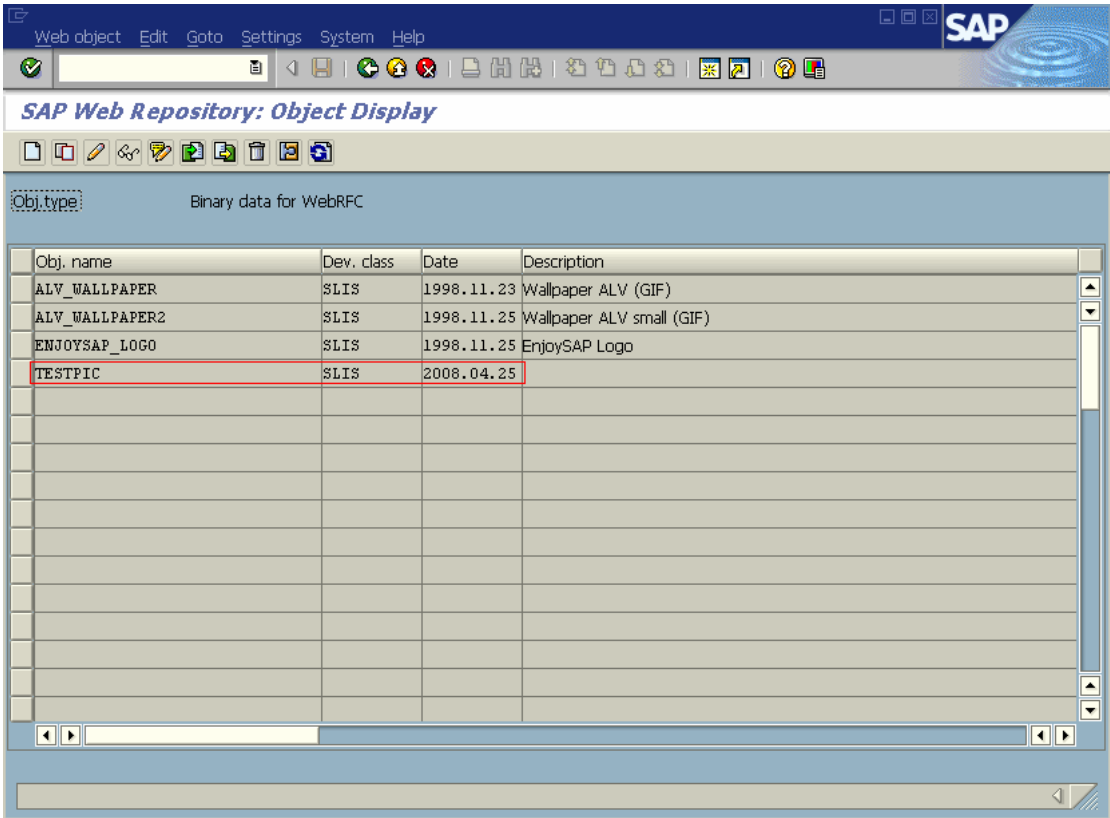


图 6-71

上传成功，屏幕显示新产生的对象行，如图 6-72 所示。

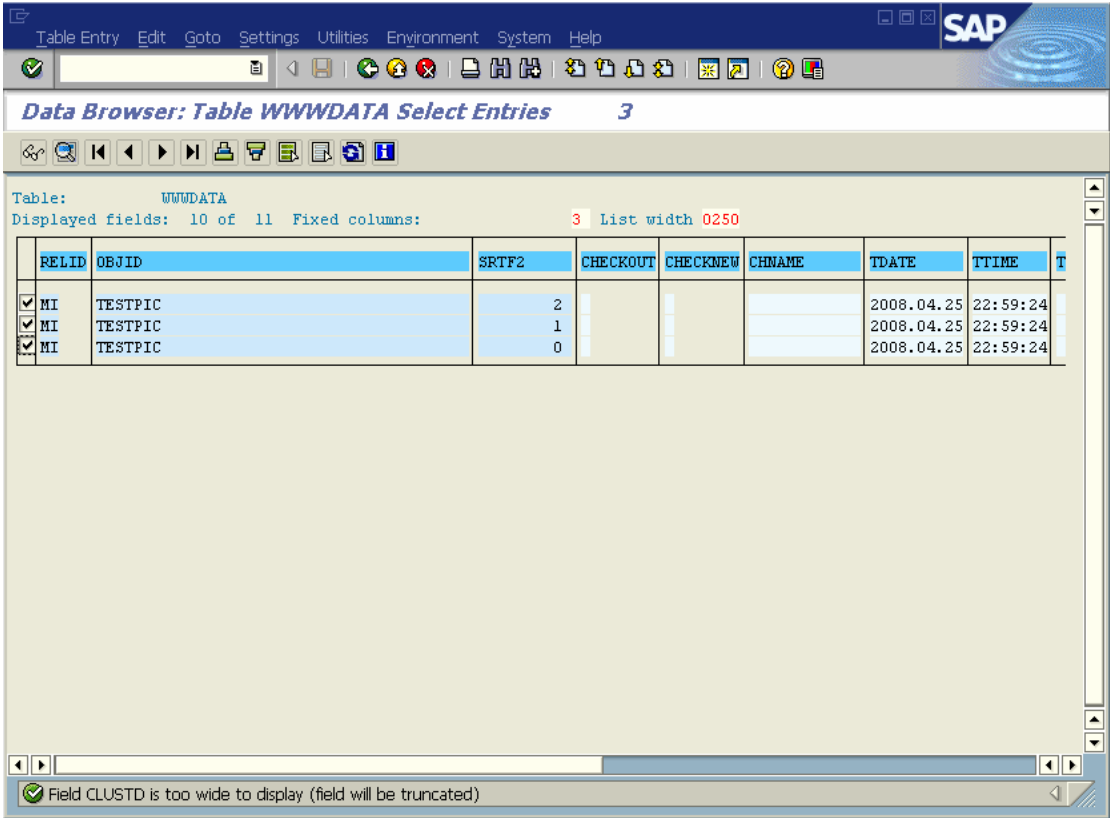


The screenshot shows the 'SAP Web Repository: Object Display' window. It contains a table with the following data:

Obj. name	Dev. class	Date	Description
ALV_WALLPAPER	SLIS	1998.11.23	Wallpaper ALV (GIF)
ALV_WALLPAPER2	SLIS	1998.11.25	Wallpaper ALV small (GIF)
ENJOYSAP_LOGO	SLIS	1998.11.25	EnjoySAP Logo
TESTPIC	SLIS	2008.04.25	

图 6-72

相关信息在数据表“WWWDATA”可以查询到，如图 6-73 所示。



The screenshot shows the 'Data Browser: Table WWWDATA Select Entries' window. It displays a table with the following data:

RELID	OBJID	SRTF2	CHECKOUT	CHECKNEW	CHNAME	TDATE	TTIME	T
<input checked="" type="checkbox"/> MI	TESTPIC	2				2008.04.25	22:59:24	
<input checked="" type="checkbox"/> MI	TESTPIC	1				2008.04.25	22:59:24	
<input checked="" type="checkbox"/> MI	TESTPIC	0				2008.04.25	22:59:24	

Field CLUSTD is too wide to display (field will be truncated)

图 6-73

6.10.2 屏幕设计定义对象

在屏幕设计界面添加定制控制对象和“退出”按钮，如图 6-74 所示。

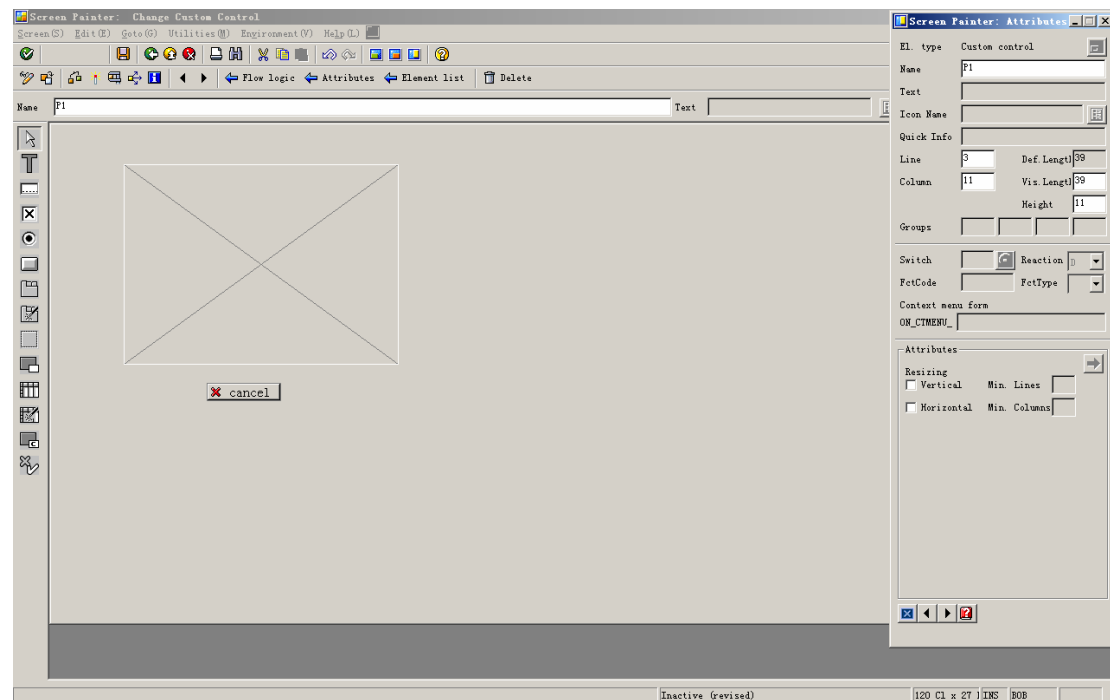


图 6-74

6.10.3 图片显示程序

程序：

```
REPORT YTEST17 .
```

```
DATA:OK_CODE TYPE SY-UCOMM,
```

```
      SAVE_OK TYPE SY-UCOMM.
```

**定义控制和图像对象定义*

```
DATA CONTAINER TYPE REF TO CL_GUI_CUSTOM_CONTAINER.
```

```
DATA PICTURE TYPE REF TO CL_GUI_PICTURE.
```

**初始标志字段定义*

```
DATA INIT.
```

```
CALL SCREEN 100.
```

```
*&-----*
```

```
*&      Module  USER_COMMAND_0100  INPUT
```

```
*&-----*
```

```
*      text
```

```
*-----*
```

```
MODULE USER_COMMAND_0100 INPUT.
```

**用户交互*

```
      SAVE_OK = OK_CODE.
```

```

    CLEAR OK_CODE.
    CASE SAVE_OK.
        WHEN 'EXIT'.
            LEAVE PROGRAM.
    ENDCASE.
ENDMODULE.                                " USER_COMMAND_0100  INPUT
*&-----*
*&      Module  STATUS_0100  OUTPUT
*&-----*
* 定义状态条、包括菜单、工具条按钮、系统按钮等
*-----*
MODULE STATUS_0100 OUTPUT.
    SET PF-STATUS 'STATUS1'.
*如果还未创建定制对象
    IF INIT IS INITIAL.
*图片地址字段定义
        DATA URL(255).
        CLEAR URL.
*创建对象
        CREATE OBJECT CONTAINER
            EXPORTING CONTAINER_NAME = 'P1'.

        CREATE OBJECT PICTURE
            EXPORTING PARENT = CONTAINER
            EXCEPTIONS ERROR = 1.
        IF SY-SUBRC NE 0.
            ENDIF.
*定义通用函数取得图片地址
        PERFORM LOAD_PIC_DB CHANGING URL.
*显示图片
        CALL METHOD PICTURE->LOAD_PICTURE_FROM_URL
            EXPORTING
                URL = URL.
        IF SY-SUBRC NE 0.
            ENDIF.
*置标志位已初始
        INIT = 'X'.

    ENDIF.
ENDMODULE.                                " STATUS_0100  OUTPUT
*&-----*
*&      Form  LOAD_PIC_DB
*&-----*
* 获取图片地址通用函数

```

```

*-----*
*      <--P_URL   text
*-----*
FORM LOAD PIC_DB   CHANGING P_URL.
  DATA QUERY_TABLE LIKE W3QUERY OCCURS 1 WITH HEADER LINE.
  DATA HTML_TABLE LIKE W3HTML OCCURS 1.
  DATA RETURN_CODE LIKE W3PARAM-RET_CODE.
  DATA CONTENT_TYPE LIKE W3PARAM-CONT_TYPE.
  DATA CONTENT_LENGTH LIKE W3PARAM-CONT_LEN.
  DATA PIC_DATA LIKE W3MIME OCCURS 0.
  DATA PIC_SIZE TYPE I.

  REFRESH QUERY_TABLE.
*查询对象数据
  QUERY_TABLE-NAME = '_OBJECT_ID'.
*图片对象名称
  QUERY_TABLE-VALUE = 'TESTPIC'.
  APPEND QUERY_TABLE.
*读取WWWDATA表, 取得图片对象信息
  CALL FUNCTION 'WWW_GET_MIME_OBJECT'
    TABLES
      QUERY_STRING      = QUERY_TABLE
      HTML              = HTML_TABLE
      MIME              = PIC_DATA
    CHANGING
      RETURN_CODE      = RETURN_CODE
      CONTENT_TYPE     = CONTENT_TYPE
      CONTENT_LENGTH   = CONTENT_LENGTH
    EXCEPTIONS
      OBJECT_NOT_FOUND = 1
      PARAMETER_NOT_FOUND = 2
      OTHERS           = 3.
  IF SY-SUBRC EQ 0.
    PIC_SIZE = CONTENT_LENGTH.
  ENDIF.
*创建图片URL地址
  CALL FUNCTION 'DP_CREATE_URL'
    EXPORTING
      TYPE              = 'image'
      SUBTYPE           = CNDP_SAP_TAB_UNKNOWN
      SIZE              = PIC_SIZE
*  DATE                =
*  TIME                =
*  DESCRIPTION         =

```

```

        LIFETIME                = CNDP_LIFETIME_TRANSACTION
*   CACHEABLE                  =
    TABLES
        DATA                  = PIC_DATA
*   FIELDS                    =
*   PROPERTIES                =
    CHANGING
        URL                    = URL
    EXCEPTIONS
        DP_INVALID_PARAMETER   = 1
        DP_ERROR_PUT_TABLE     = 2
        DP_ERROR_GENERAL       = 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.

ENDFORM.                    " LOAD_PIC_DB

```

6.10.4 程序输出

程序运行输出结果如图 6-75 所示。

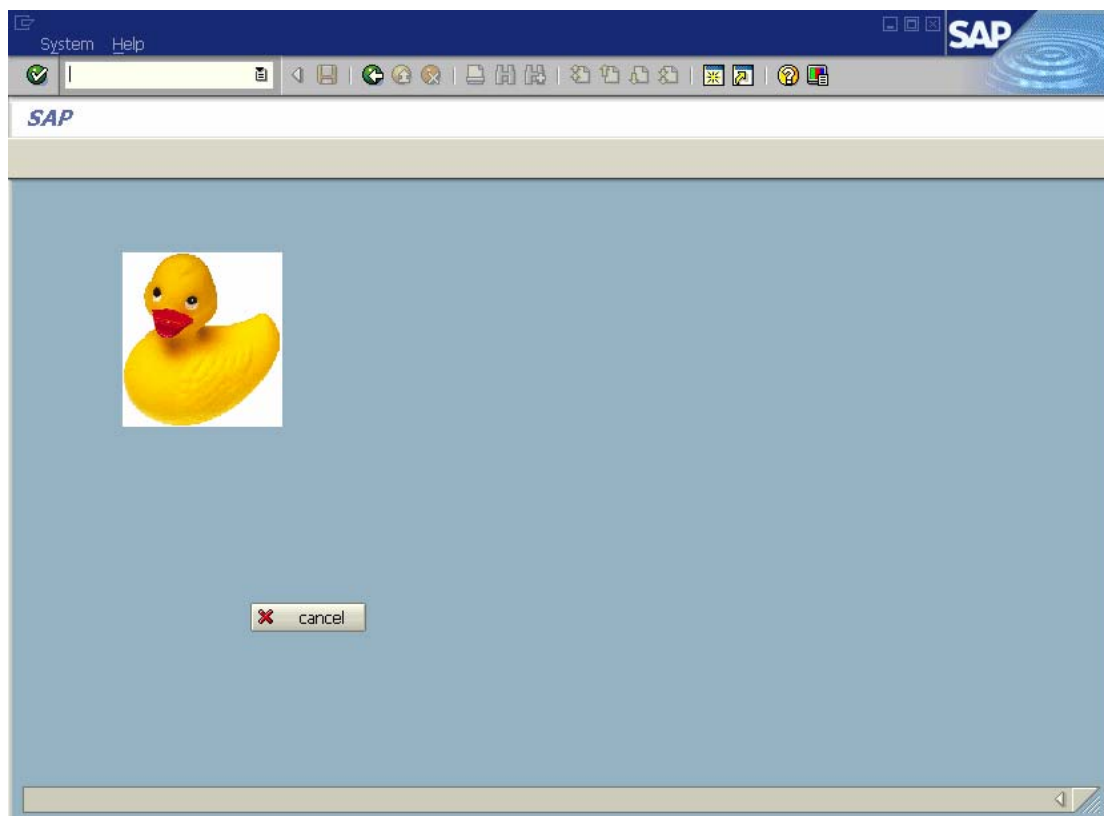


图 6-75

在此基础上, 可以添加图片双击等事件。

6.11 通过定制控制文本编辑器

在 SAP 业务系统中, 大量用到文本编辑器, 用文本编辑器处理内表, 所以本书在屏幕上创建文本编辑器, 输入相关内容后, 在屏幕退出时输出编辑器内容。

6.11.1 屏幕设计定义对象

在屏幕设计界面定义控制对象, 退出按钮。

6.11.2 文本编辑器程序

程序:

```
REPORT YTEST18.
DATA: OK_CODE TYPE SY-UCOMM,
      SAVE_OK TYPE SY-UCOMM.
*初始标志字段定义
*定制控制和编辑对象定义
DATA: INIT,
      CONTAINER TYPE REF TO CL_GUI_CUSTOM_CONTAINER,
      EDITOR TYPE REF TO CL_GUI_TEXTEDIT.

DATA: M1(256) TYPE C OCCURS 0,
      LINE(256) TYPE C.
*输入初始值
LINE = '请输入'.
APPEND LINE TO M1.

CALL SCREEN 100.

REFRESH M1.

CALL METHOD EDITOR->GET_TEXT_AS_R3TABLE
IMPORTING
TABLE = M1.
*逐行输出
LOOP AT M1 INTO LINE.
WRITE / LINE.
ENDLOOP.
*&-----*
```

```

*&      Module  USER_COMMAND_0100  INPUT
*&-----*
*      text
*-----*
MODULE USER_COMMAND_0100 INPUT.
    SAVE_OK = OK_CODE.
    CLEAR OK_CODE.
    CASE SAVE_OK.
        WHEN 'EXIT'.
            LEAVE TO SCREEN 0.
    ENDCASE.
ENDMODULE.                                " USER_COMMAND_0100  INPUT
*&-----*
*&      Module  STATUS_0100  OUTPUT
*&-----*
*      text
*-----*
MODULE STATUS_0100 OUTPUT.
    SET PF-STATUS 'STATUS1'.
*如果还未创建定制对象
    IF INIT IS INITIAL.
        INIT = 'X'.

        CREATE OBJECT: CONTAINER EXPORTING CONTAINER_NAME = 'P1'.

        CREATE OBJECT EDITOR
            EXPORTING
                PARENT = CONTAINER
                WORDWRAP_MODE = CL_GUI_TEXTEDIT=>WORDWRAP_AT_FIXED_POSITION
                WORDWRAP_POSITION = 256
                WORDWRAP_TO_LINEBREAK_MODE = CL_GUI_TEXTEDIT=>TRUE.
    ENDIF.
*读入初始数据
    CALL METHOD EDITOR->SET_TEXT_AS_R3TABLE
        EXPORTING
            TABLE = M1.

ENDMODULE.                                " STATUS_0100  OUTPUT

```

6.11.3 输出

程序运行后, 屏幕显示文本编辑器, 在编辑框内输入文字, 如图 6-76 所示。

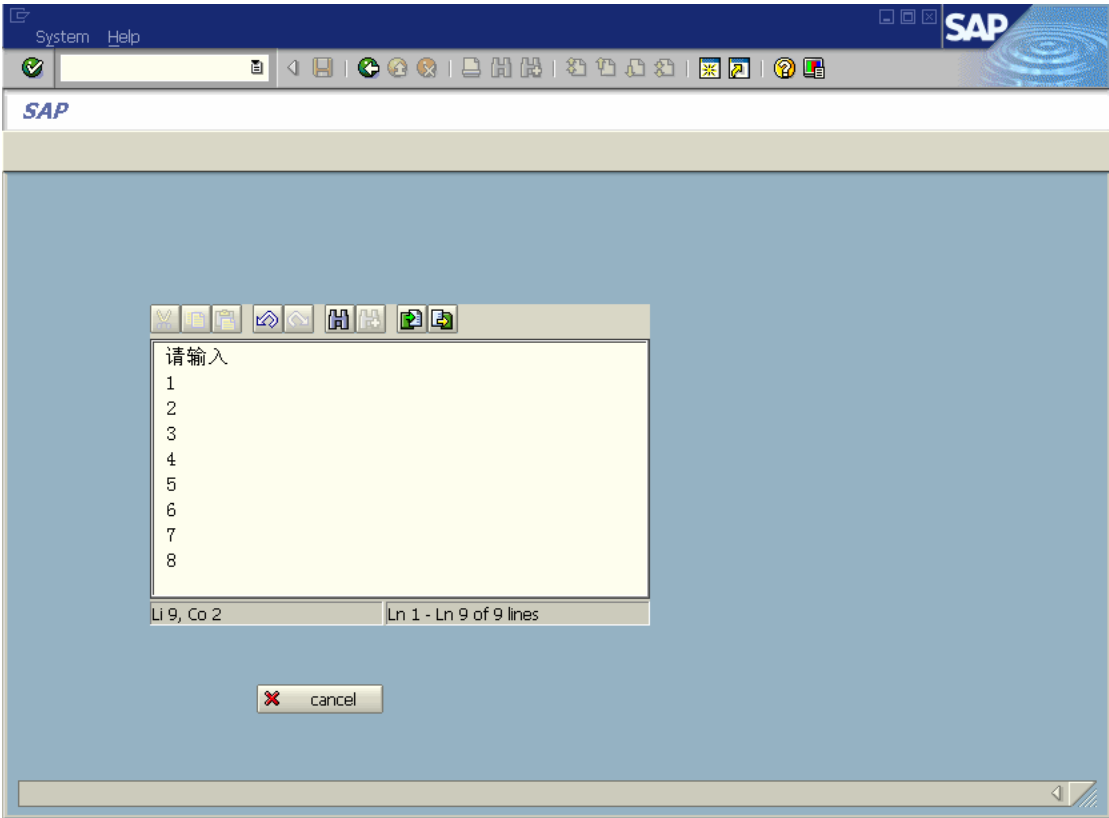


图 6-76

单击“cancel”按钮，退出文本编辑器屏幕，输出内表记录，如图 6-77 所示。

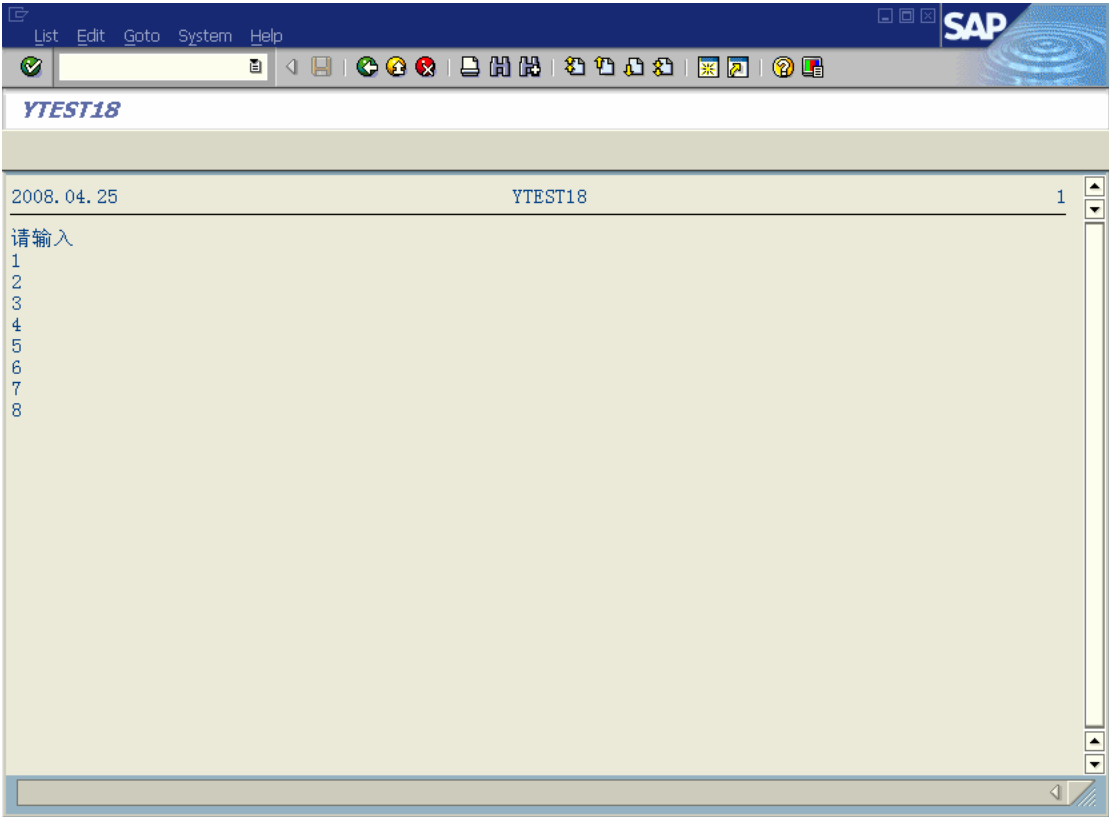


图 6-77

6.12 列表和屏幕相互调用

6.12.1 从屏幕输入条件，列表输出数据

以 SPFLI 表（航空表）为例，建立一个输入字段和一个查询按钮的屏幕，当单击查询按钮时，根据该输入字段检索，将数据通过列表输出。

首先建立一个程序“YTEST019”，然后建立窗口，输入 TCODE: SE51，输入程序名和屏幕编号，单击“创建”按钮，如图 6-78 所示。

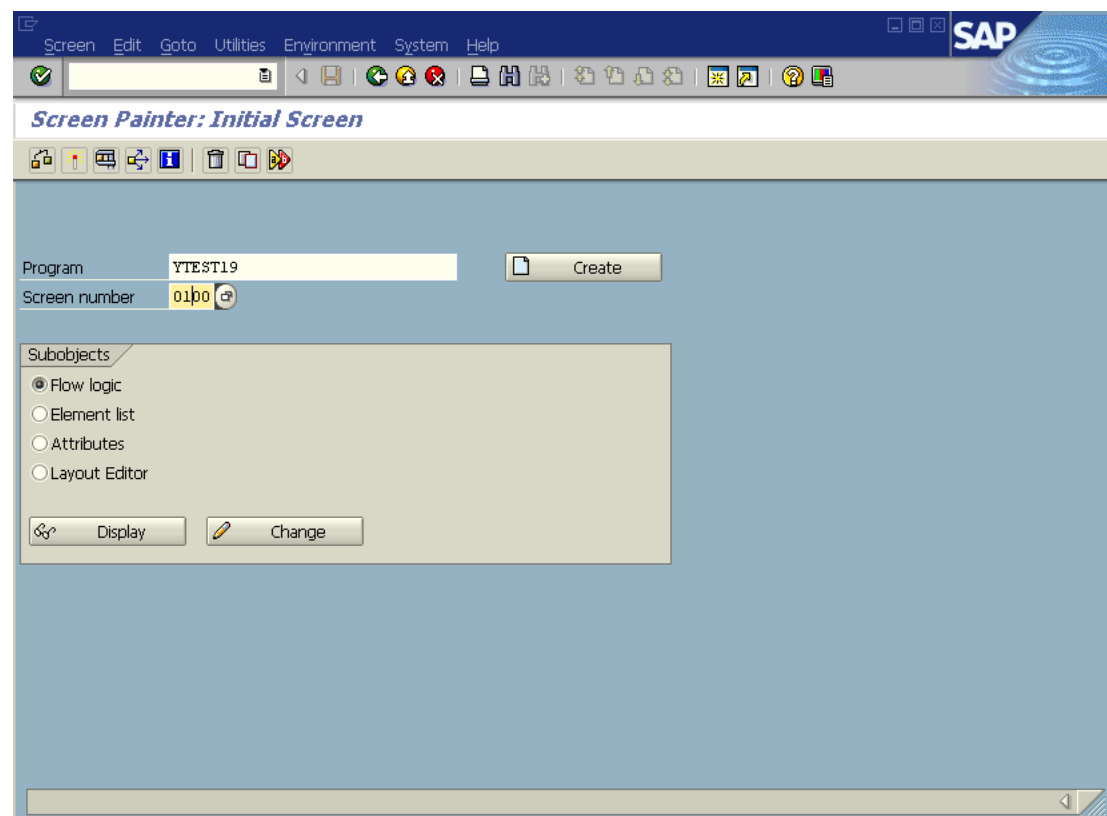



图 6-78

输入描述后保存，单击  Layout 按钮，如图 6-79 所示。

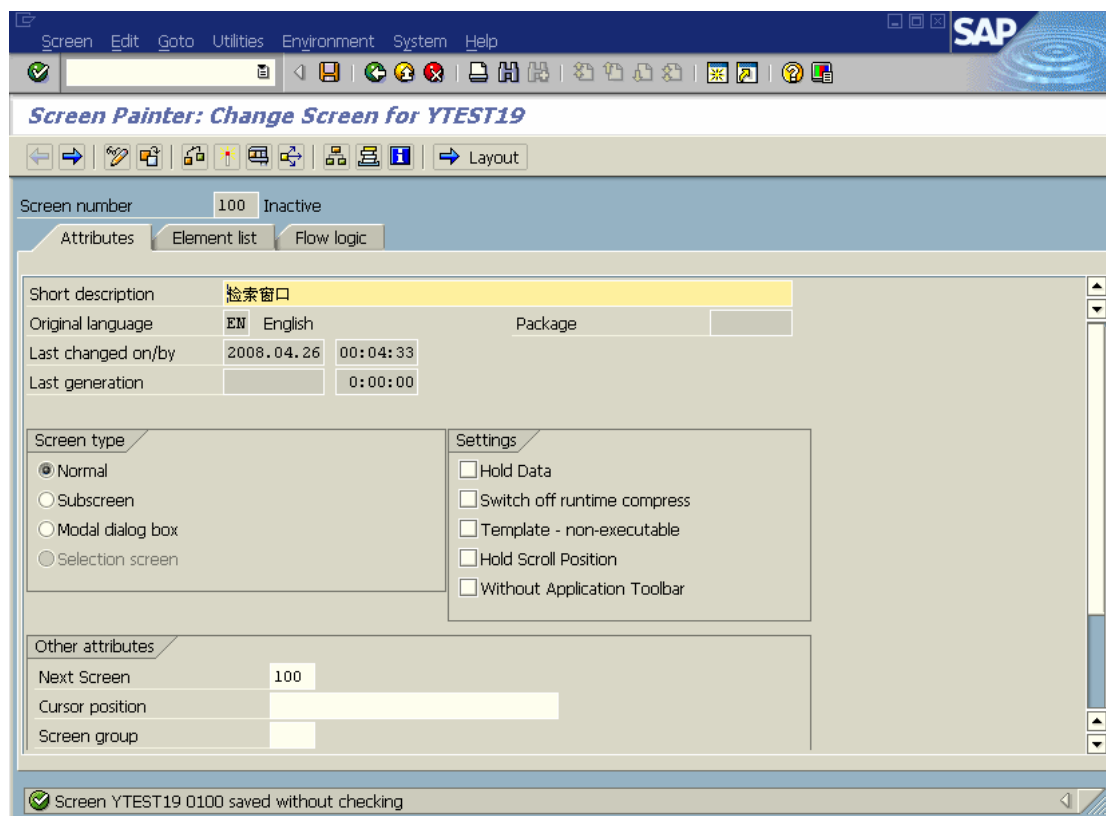


图 6-79

添加一个文本字段，并定义名称和文本，如图 6-80 所示。

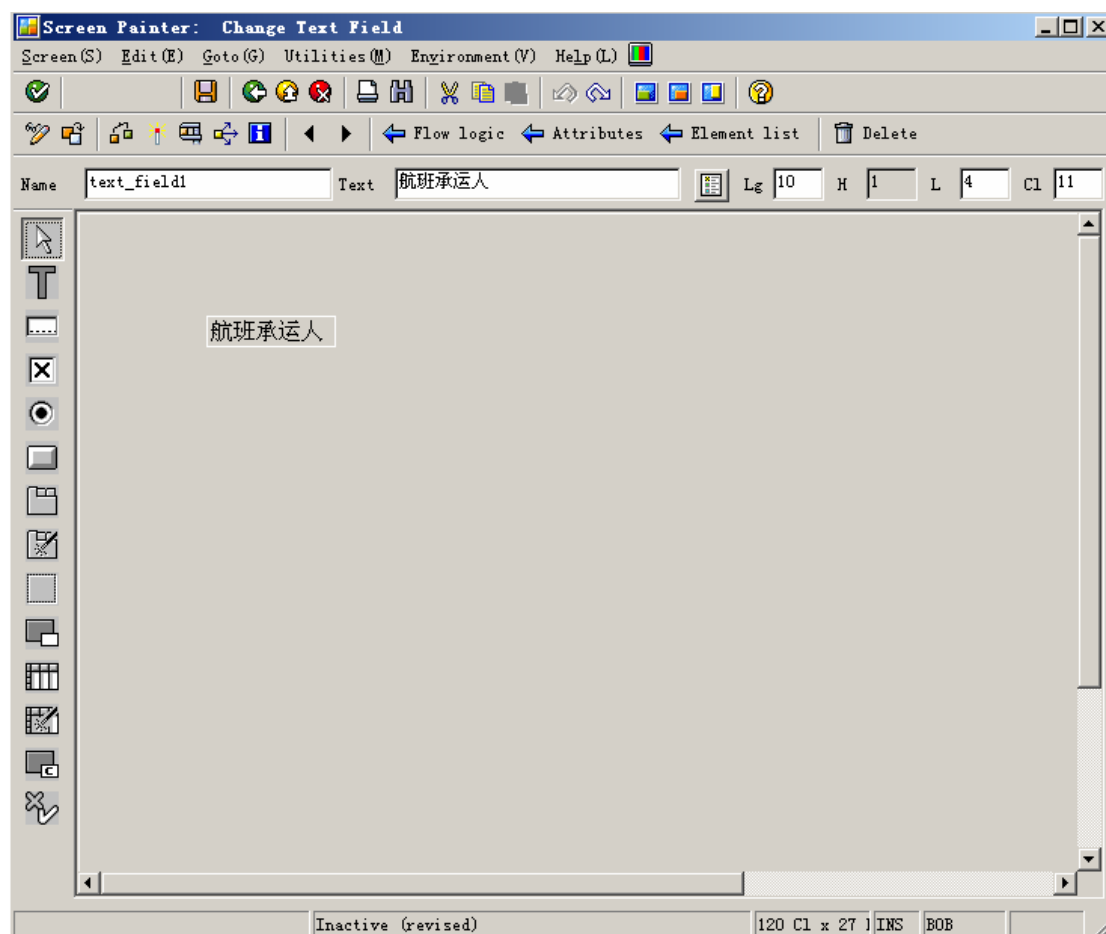


图 6-80

然后建立输入框，字段属性为输入输出字段，再添加“确认”按钮和“取消”按钮，如图 6-81 所示。

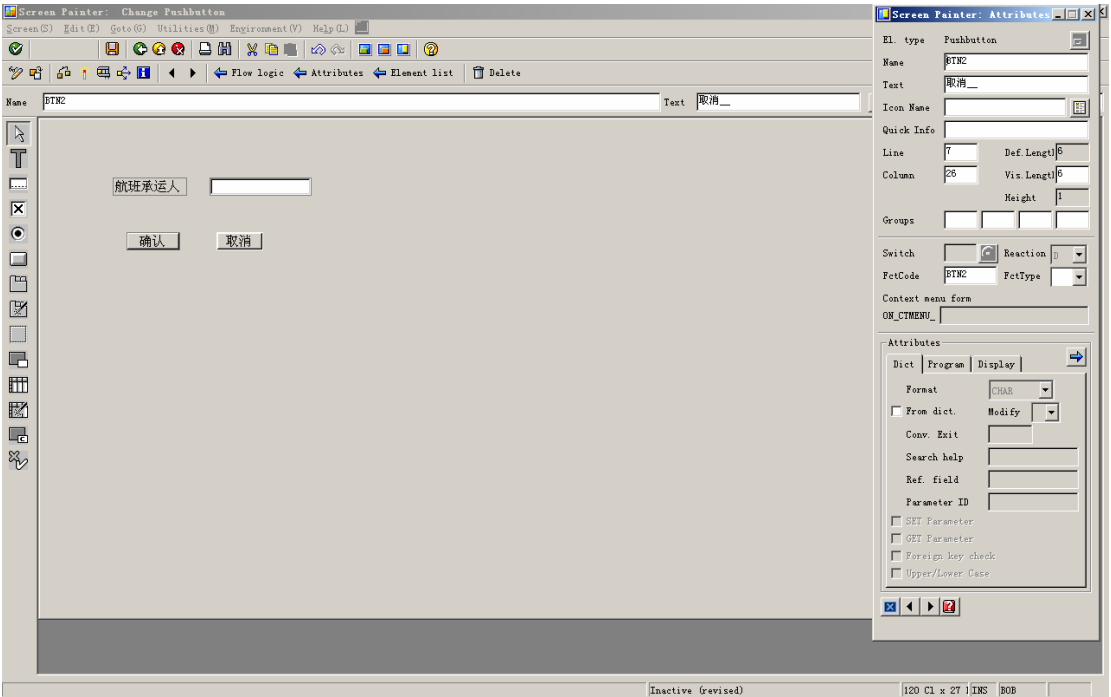


图 6-81

设计逻辑程序，如图 6-82 所示。

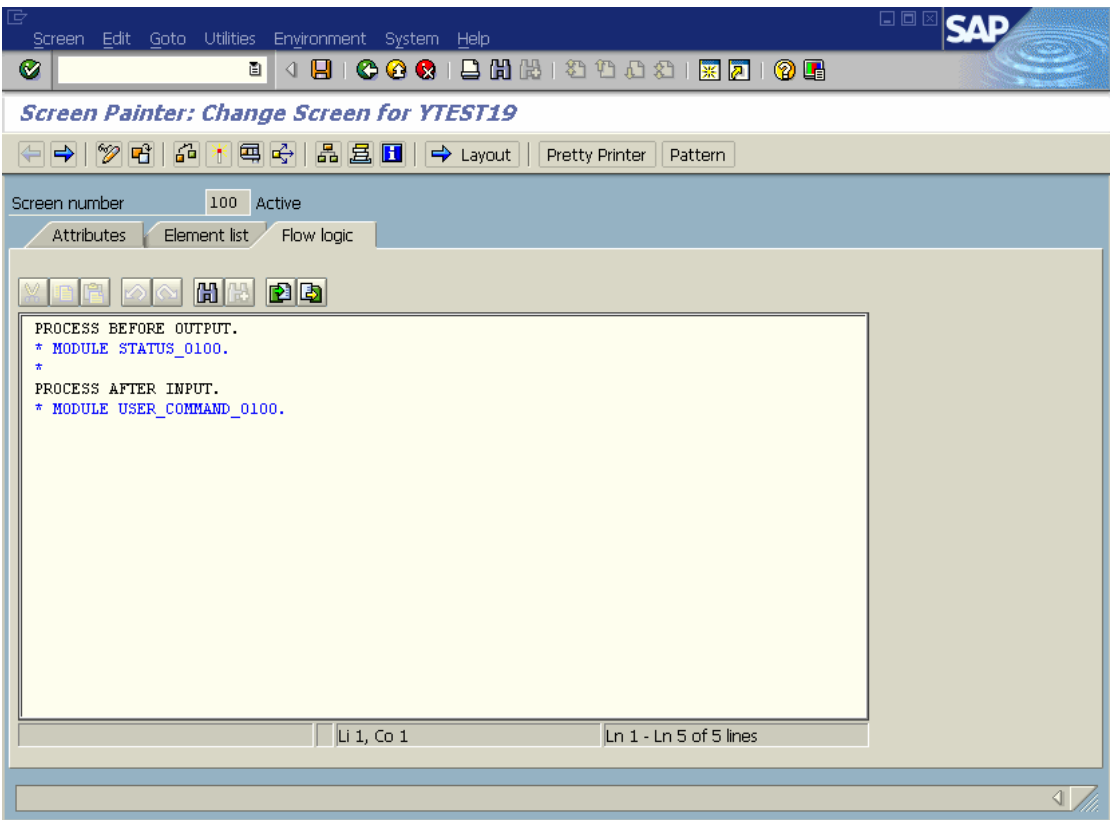


图 6-82

去掉程序“MODULE USER_COMMAND_0100.”前的符号“*”，使该行生效，加入程

序 “MODULE CANCEL AT EXIT-COMMAND.” 表示单击退出按钮时执行 CANCEL 模块。
切换到元素清单页面，加入返回变量 “OK_CODE”，如图 6-83 所示。

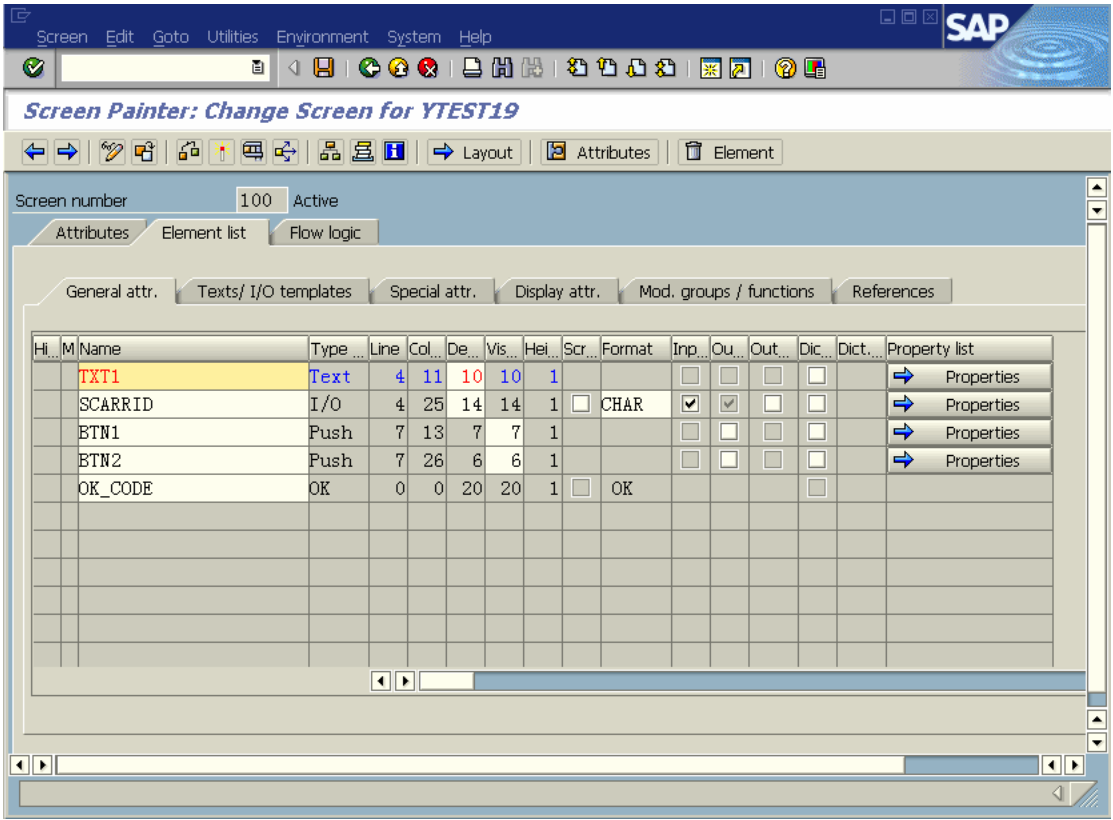


图 6-83

主程序:

```
REPORT YTEST19.
DATA: OK_CODE TYPE SY-UCOMM,
      SAVE_OK TYPE SY-UCOMM,
      SCARRID(20) TYPE C.
TABLES SPFLI.

CALL SCREEN 100.
*&-----*
*&      Module CANCEL INPUT
*&-----*
*      text
*-----*
MODULE CANCEL INPUT.
  LEAVE PROGRAM.
ENDMODULE.              " CANCEL INPUT
*&-----*
*&      Module USER_COMMAND_0100 INPUT
*&-----*
*      text
*-----*
```

```
MODULE USER_COMMAND_0100 INPUT.  
  SAVE_OK = OK_CODE.  
  CLEAR OK_CODE.  
  IF SAVE_OK EQ 'BTN1'.  
    LEAVE TO LIST-PROCESSING AND RETURN TO SCREEN 0.  
    SUPPRESS DIALOG.  
    SELECT *  
      FROM SPFLI WHERE CARRID = SCARRID.  
    WRITE: / SPFLI-CARRID, SPFLI-CONNID,  
            SPFLI-CITYFROM, SPFLI-CITYTO.  
  ENDSELECT.  
ENDIF.  
WRITE: / ' DONE'.  
ENDMODULE.                                " USER_COMMAND_0100 INPUT
```

输出结果如图 6-84 所示, 在输入框输入条件, 单击“确认”按钮。

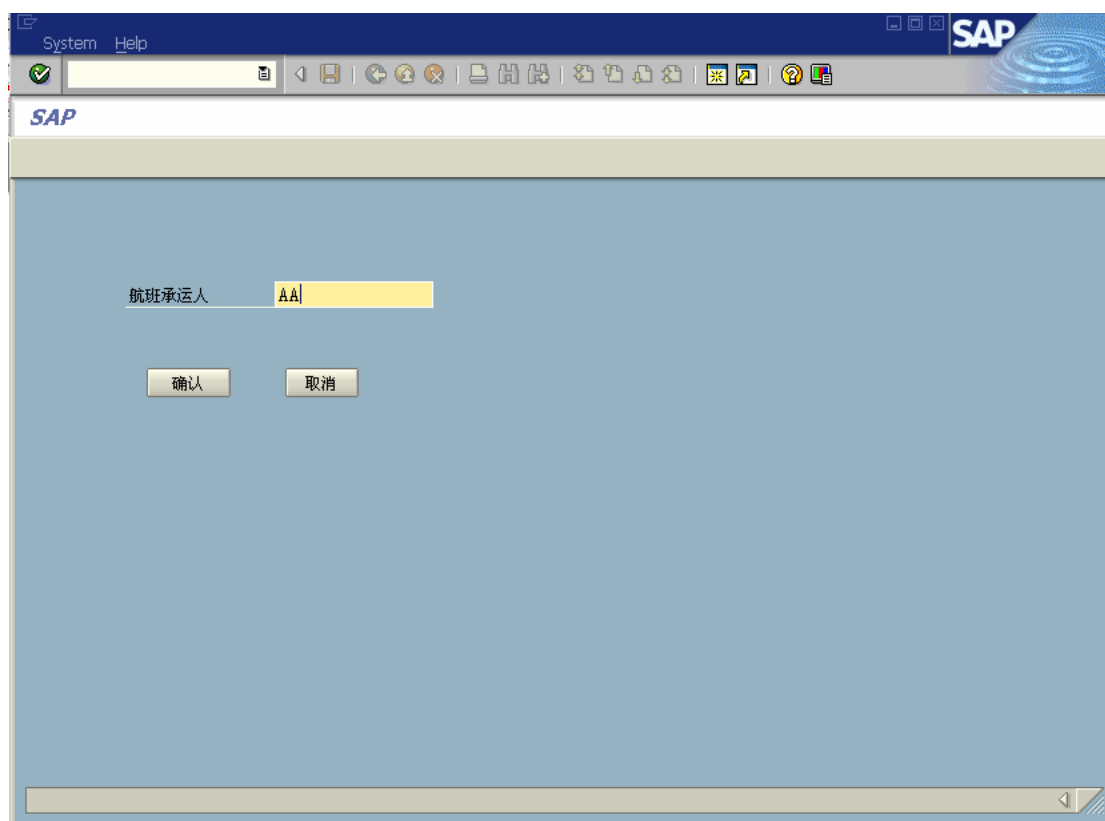


图 6-84

系统输出如图 6-85 所示。

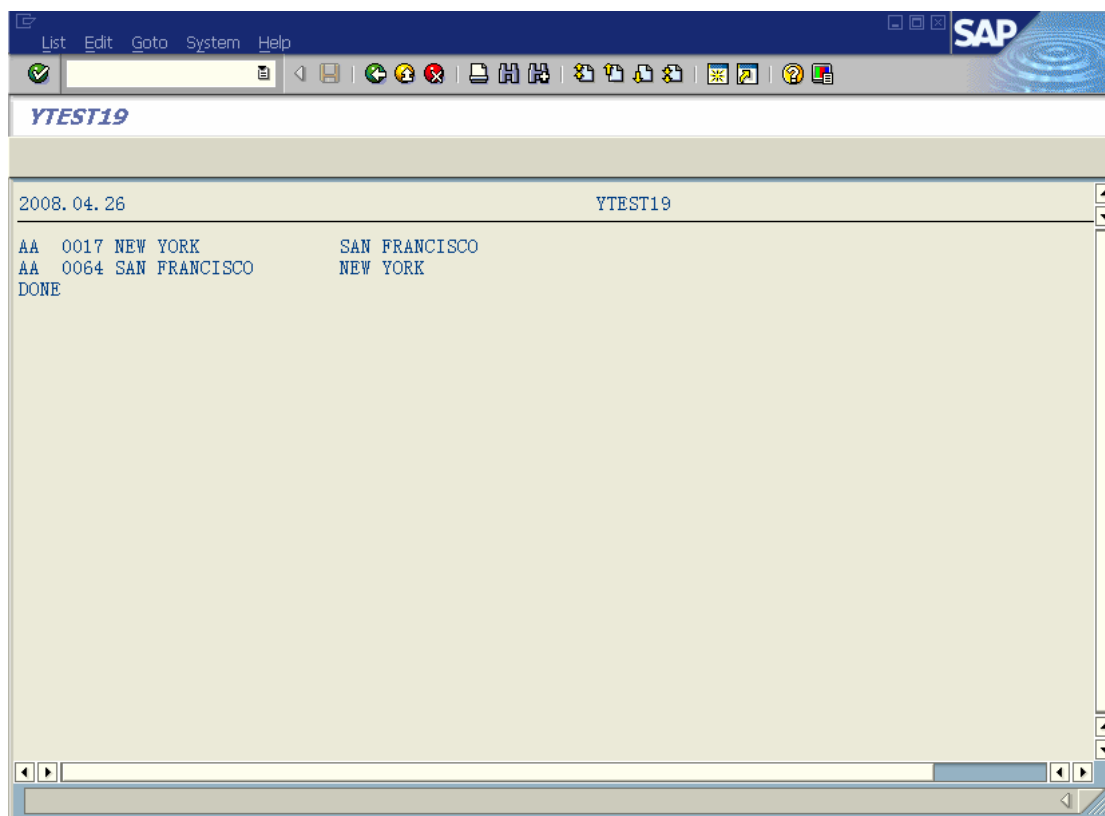


图 6-85

6.12.2 从列表调用屏幕

以 SPFLI 表（航班表）为例，在屏幕上输出列表，当双击某一行时，调用屏幕显示数据明细。

首先建立程序 YTEST20，然后建立显示屏幕。

输入 TCODE: SE51，建立编号 100 的屏幕，单击“字典字段”按钮，输入航班表后，屏幕显示表字段，选择字段，确认，如图 6-86 所示。

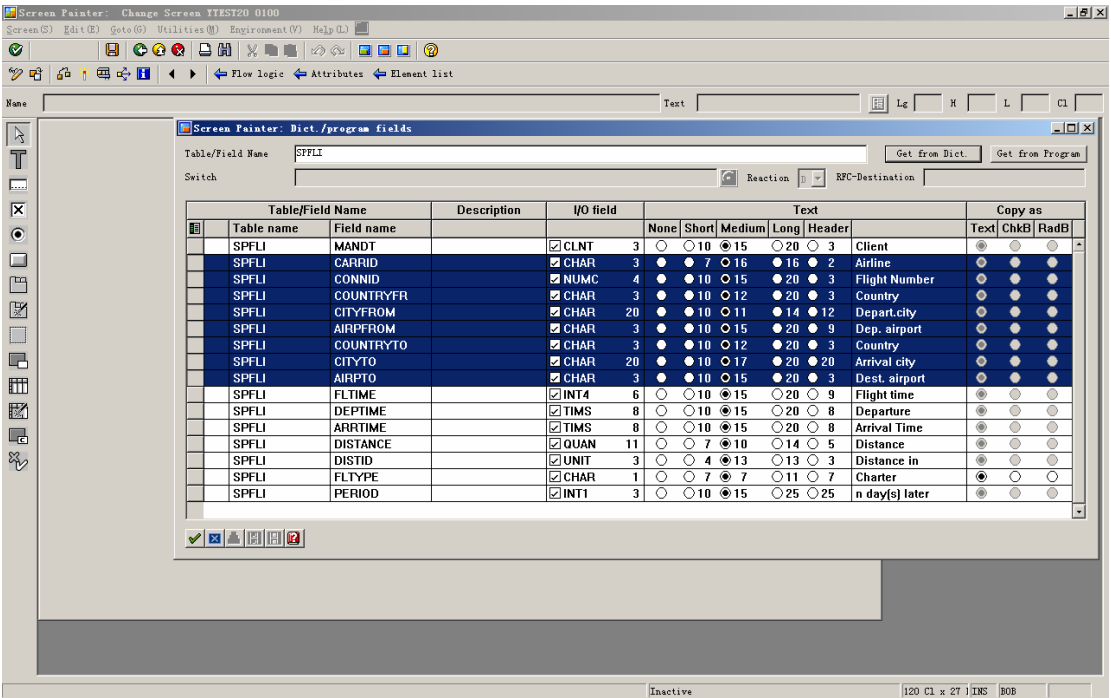


图 6-86

选择输出位置，单击后，系统自动创建输入文本和输入字段，添加退出按钮，如图 6-87 所示。

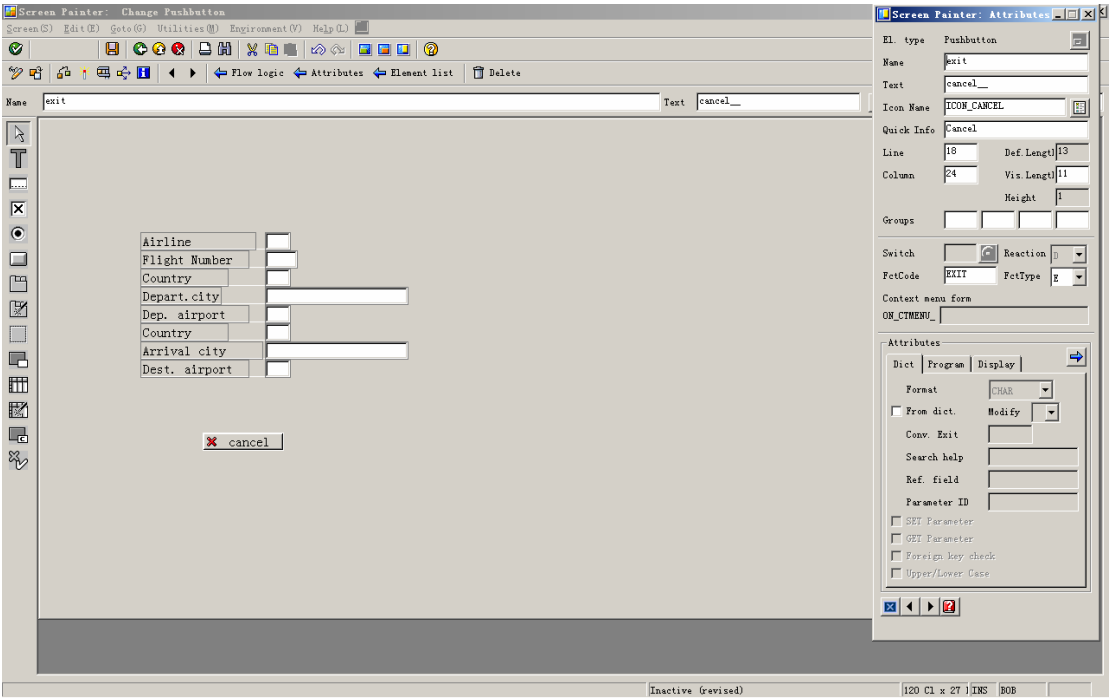



图 6-87

单击  Flow logic 按钮，定义逻辑程序，如图 6-88 所示。

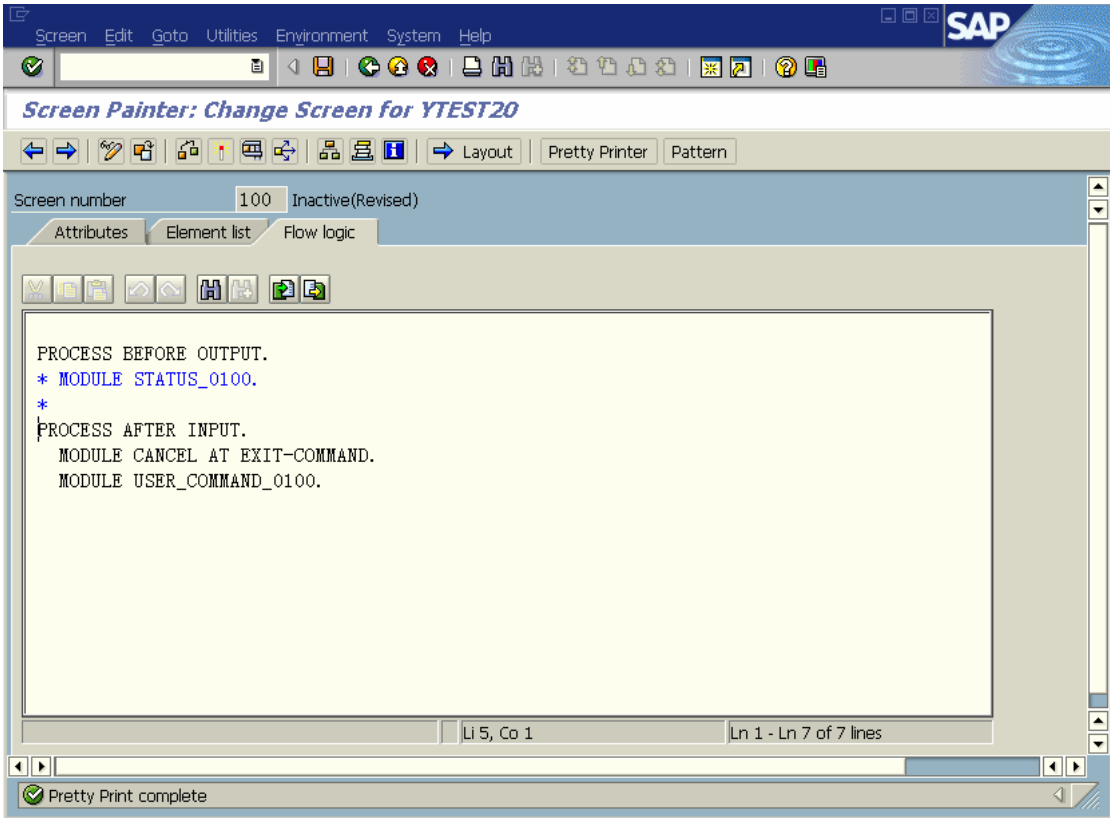


图 6-88

切换到元素清单页面，定义变量“OK_CODE”，如图 6-89 所示。

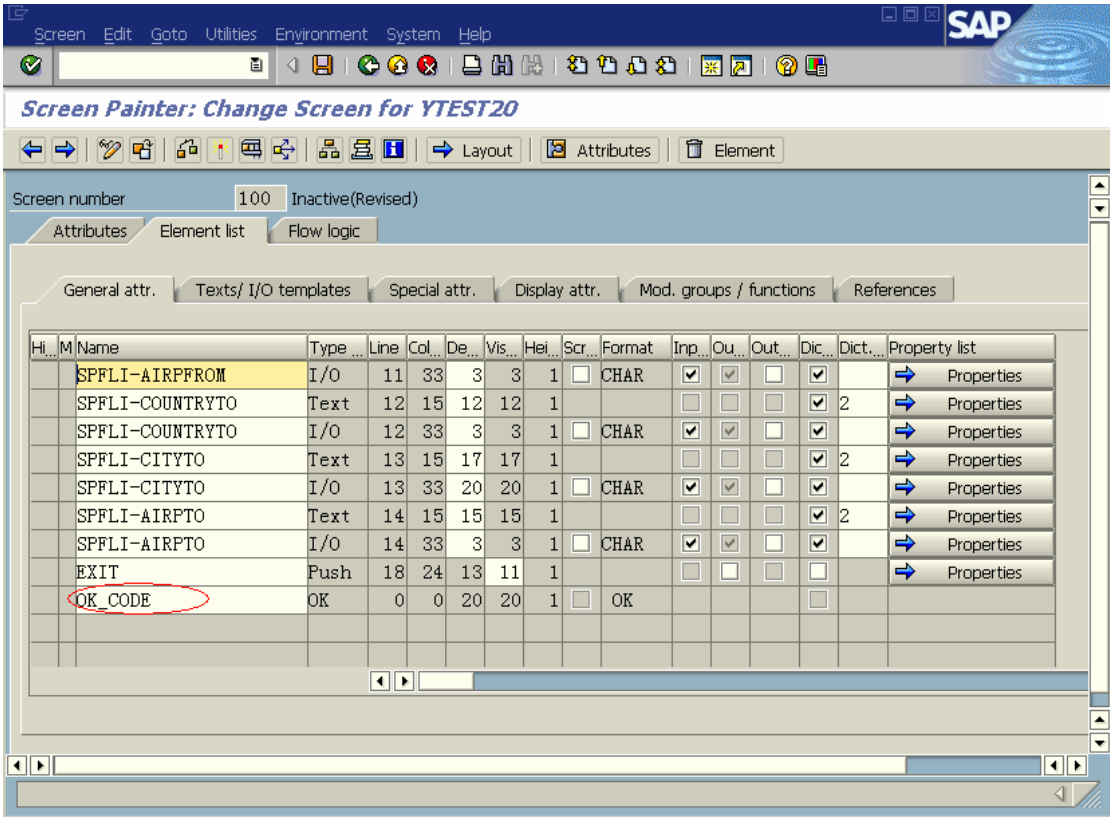


图 6-89

单击保存按钮，屏幕建立完成。

主程序：

REPORT YTEST20 .

TABLES SPFLI.

START-OF-SELECTION.

SELECT * FROM SPFLI.

WRITE: / SPFLI-CARRID, (15) SPFLI-CONNID,
(15) SPFLI-COUNTRYTO, (15) SPFLI-CITYFROM,
(15) SPFLI-AIRPFROM.

HIDE: SPFLI-CARRID, SPFLI-CONNID, SPFLI-COUNTRYFR,
SPFLI-CITYFROM, SPFLI-AIRPFROM, SPFLI-COUNTRYTO,
SPFLI-CITYTO, SPFLI-AIRPTO.

ENDSELECT.

AT LINE-SELECTION.

CHECK NOT SPFLI-CARRID IS INITIAL.

CALL SCREEN 100.

&-----

*& Module CANCEL INPUT

&-----

* text

MODULE CANCEL INPUT.

LEAVE PROGRAM.

ENDMODULE. " CANCEL INPUT

&-----

*& Module USER_COMMAND_0100 INPUT

&-----

* text

MODULE USER_COMMAND_0100 INPUT.

ENDMODULE. " USER_COMMAND_0100 INPUT

执行结果如图 6-90 所示。

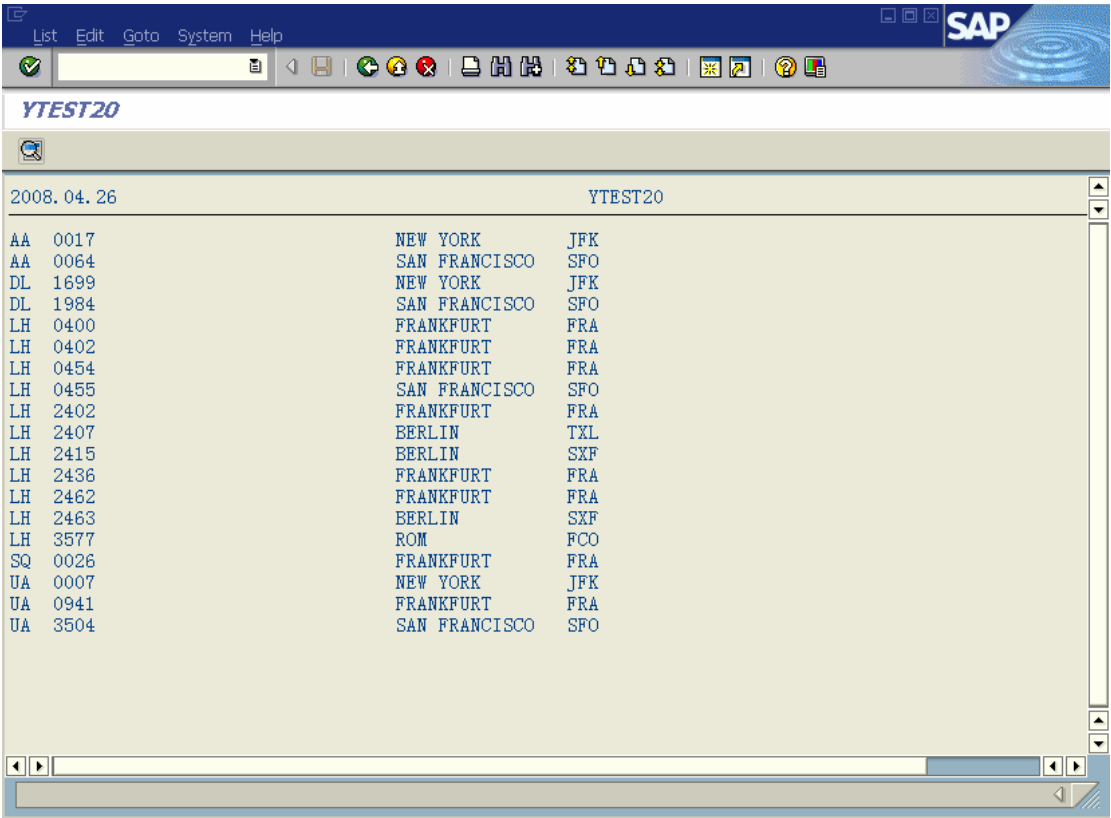


图 6-90

当选择某一行，输出结果如图 6-91 所示。

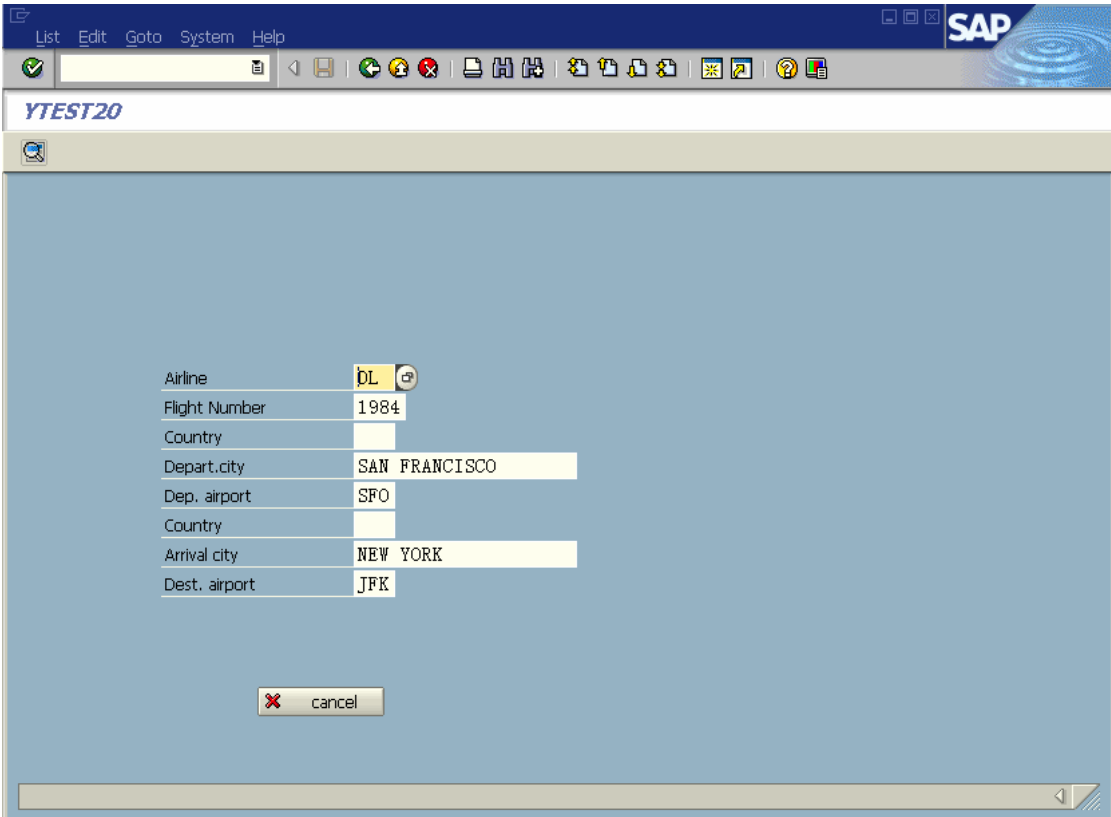


图 6-91