0708 | 0709 | ABAP

Z_PACKAGE_05010 | Z_0707_0502 | 요구사항

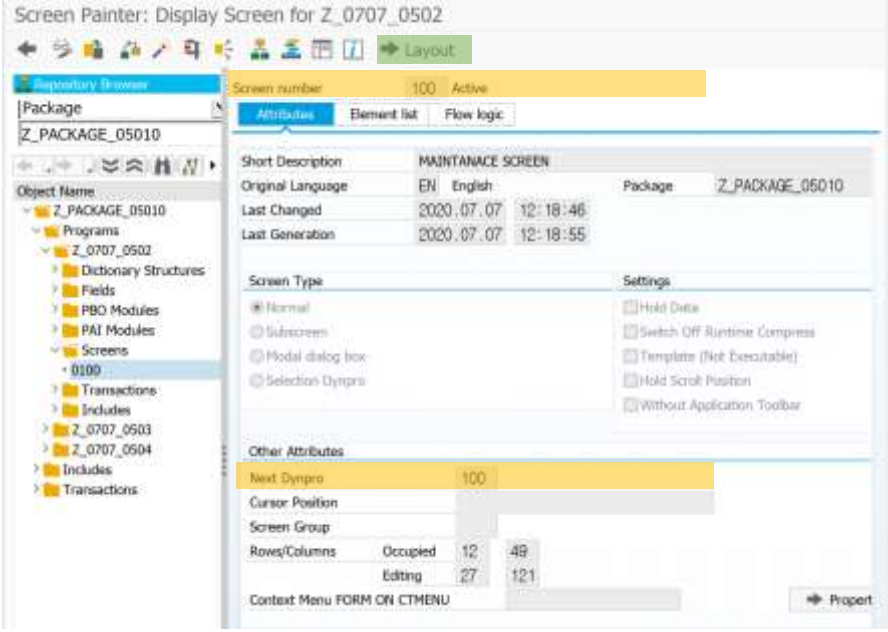
운항 일정 테이블 (SDYN_CONN) 에서 운항 일정이 존재하는 항공편에 대한 FLIGHT CONNECTION NUMBER | DATE를 입력했을 때 항공편 테이블 (SFLIGHT) 에서 AIRFARE | AIRLINE CURRENCY | PLANE TYPE | MAX. CAPACITY ECON. | OCCUPIED ECON. | TOTAL 정보를 조회하고자 한다. 사용자에게 입력 받는 AIRLINE | FLIGHT NUMBER | DATE 값은 필수 값으로 지정하며, 이를 입력했을 시 COMMAND 창에 'X'를 입력하면 프로그램이 종료되도록 구현한다.

TABLE & STR	SDYN_CONN | SFLIGHT
SCREEN	100
INPUT	AIRLINE | FLIGHT CONNECTION NUMBER | DATE
OUTPUT	AIRFARE | AIRLINE CURRENCY | PLANE TYPE | MAX. CAPACITY ECON. | OCCUPIED ECON. | TOTAL
COMMAND	X 〉〉 LEAVE PROGRAM

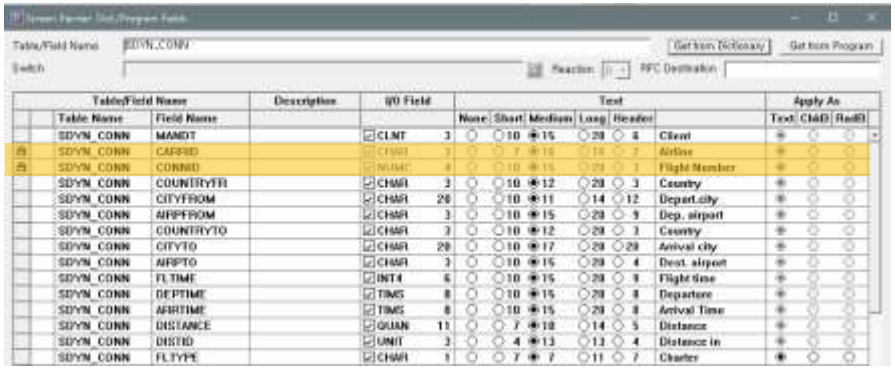
SE80 | Z_PACKAGE_05010 | Z_0707_0502 | MOD.POOL

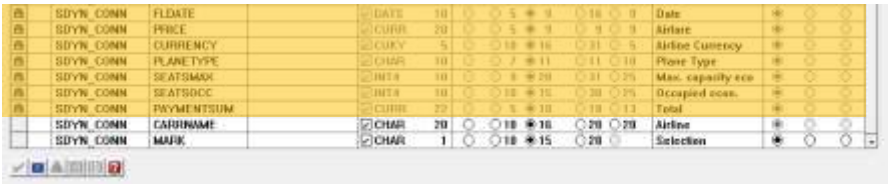
```
PROGRAM Z_0707_0502.

INCLUDE Z_0707_0502_TOP.

INCLUDE z_0707_0502_o05.

INCLUDE z_0707_0502_I05.
```

SCREEN |

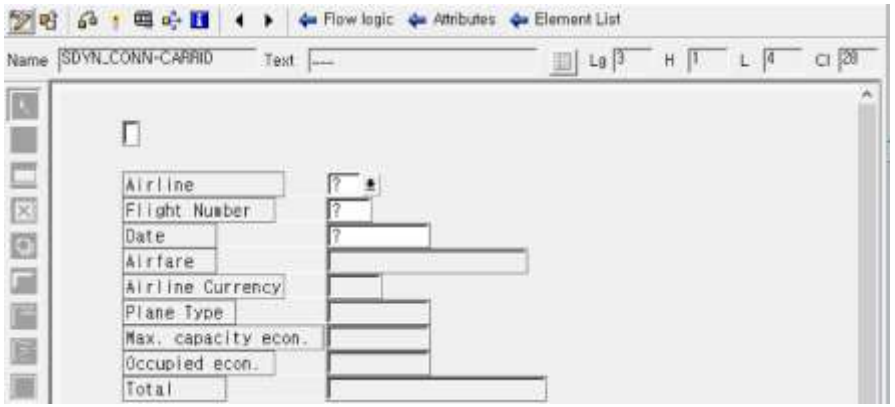


LAYOUT |

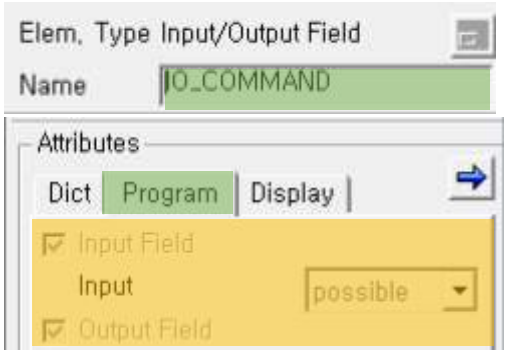




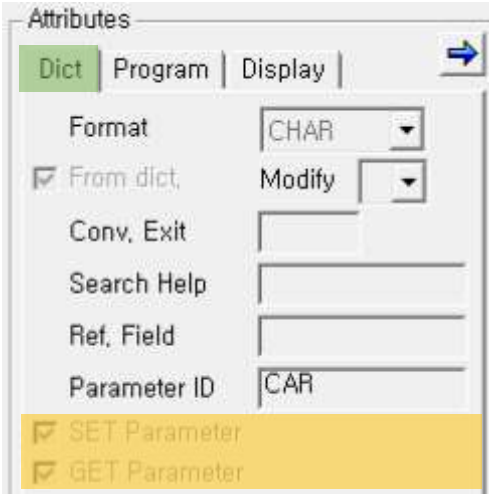
LAYOUT |

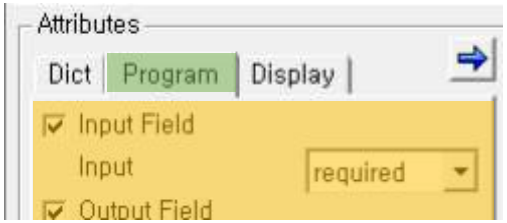


COMMAND |



INPUT |





OUTPUT |

LAYOUT | FLOW LOGIC

```
PROCESS BEFORE OUTPUT. "PBO 화면이 보여 지기 이전
 MODULE STATUS_0100.
 MODULE MOVE_TO_DYNP.
PROCESS AFTER INPUT."PAI 화면에 입력된 값을 처리하기 위한 로직
MODULE CHECK_SPFLIGHT.
MODULE USER_COMMAND_0100.
```

Z_0707_0502_TOP |

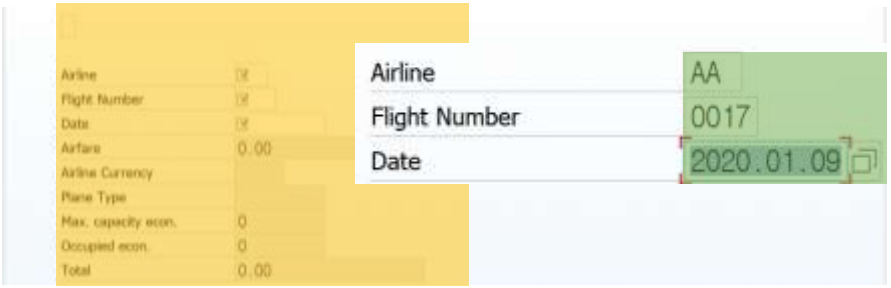
```
*DATA 선언
TABLES SDYN_CONN.
*SCREEN에서 입력 받은 값을 SELECT 해오기 위해 선언
DATA WA_SPFLI TYPE SFLIGHT.
DATA IO_COMMAND.
```

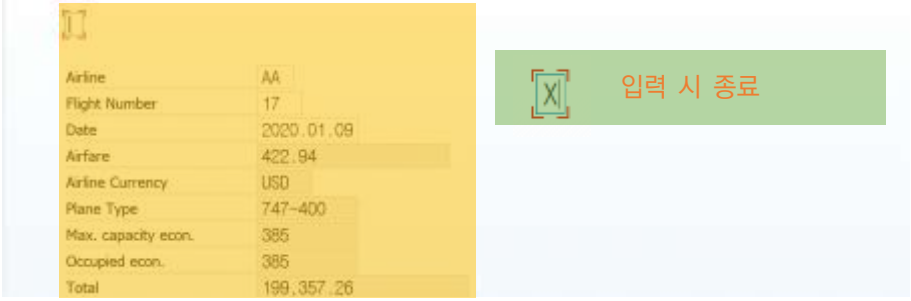
Z_0707_0502_O05 |

```
MODULE STATUS_0100 OUTPUT.

ENDMODULE.
*&---------------------------------------------------------
MODULE MOVE_TO_DYNP OUTPUT.
            MOVE-CORRESPONDING WA_SPFLI TO SDYN_CONN.
ENDMODULE.
```

Z_0707_0502_I05 |

```
MODULE USER_COMMAND_0100 INPUT.
      IF IO_COMMAND = 'X'. "COMMAND에 X를 입력할 시 조건.
         LEAVE PROGRAM. "프로그램을 종료
         "LEAVE TO SCREEN 0 .
      ENDIF.
ENDMODULE.
*&---------------------------------------------------------
MODULE CHECK_SPFLIGHT INPUT.
  SELECT SINGLE *
                  FROM SFLIGHT
                  INTO WA_SPFLI
                  WHERE CARRID = SDYN_CONN-CARRID
                      AND CONNID = SDYN_CONN-CONNID
                      AND FLDATE = SDYN_CONN-FLDATE.

                  CHECK SY-SUBRC <> 0 .
               IF SY-SUBRC <> 0.
                  CLEAR WA_SPFLI.
                  MESSAGE I007(ZBC4100) .
               ENDIF.
ENDMODULE.
```





Z_PACKAGE_05011 | Z_0708_0501 | 요구사항

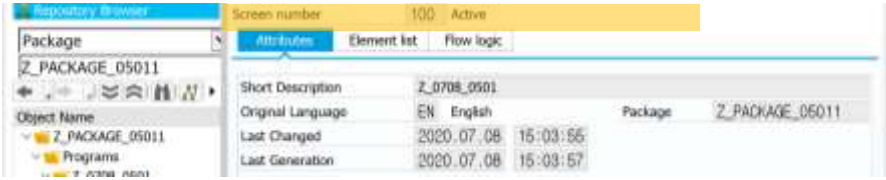
항공사 테이블 (SCARR) 에서 항공사 코드 | AIRLINE CODE를 입력하면 AIRLINE | AIRLINE CURRENCY | URL 을 조회하고자 한다. 화면은 사용자가 항공사 코드를 입력 받는 화면과 조회 화면으로 구성되어 있으며 상단 STATUS BAR의 BACK 버튼을 누르면 이전 화면으로 돌아가도록 구현한다.

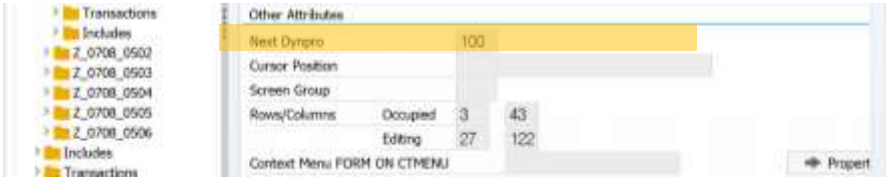
TABLE	SCARR
SCREEN	100 150
INPUT	AIRLINE CODE
OUTPUT	AIRLINE | AIRLINE CURRENCY | URL
STATUS	BACK
BUTTON	SEARCH (FCTCODE = DISP)

SE80 | MOD. POOL

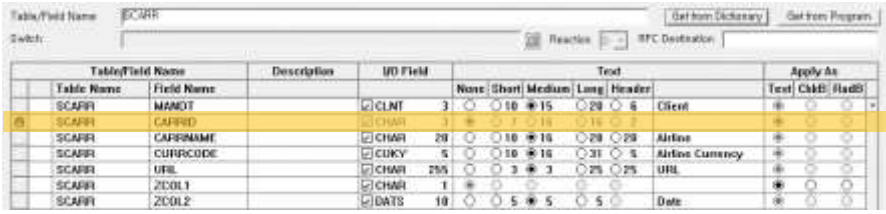
```
INCLUDE Z_0708_0501_TOP   .  " Global Data
INCLUDE Z_0708_0501_O01   .  " PBO-Modules
INCLUDE Z_0708_0501_I01   .  " PAI-Modules
```

SCREEN | 100





LAYOUT |



INPUT |

Elem. Type Input/Output Field

Name SCARR-CARRID

Text ---

Attributes

Dict | Program | Display |

Format CHAR

From dict. Modify

Conv. Exit

Search Help

Ref. Field

Parameter ID CAR

SET Parameter

GET Parameter

Attributes

Dict | Program | Display |

Input Field

Input required

Output Field

Output Only

Possible Entries

INPUT |

Elem. Type Pushbutton

Name SEARCH

Text SEARCH

Switch Reaction D

FctCode DISP FctType

Context Menu Form

ON_CTMENU_

SCREEN | 100 | ELEMENT LIST

Attributes | Element list | Flow logic

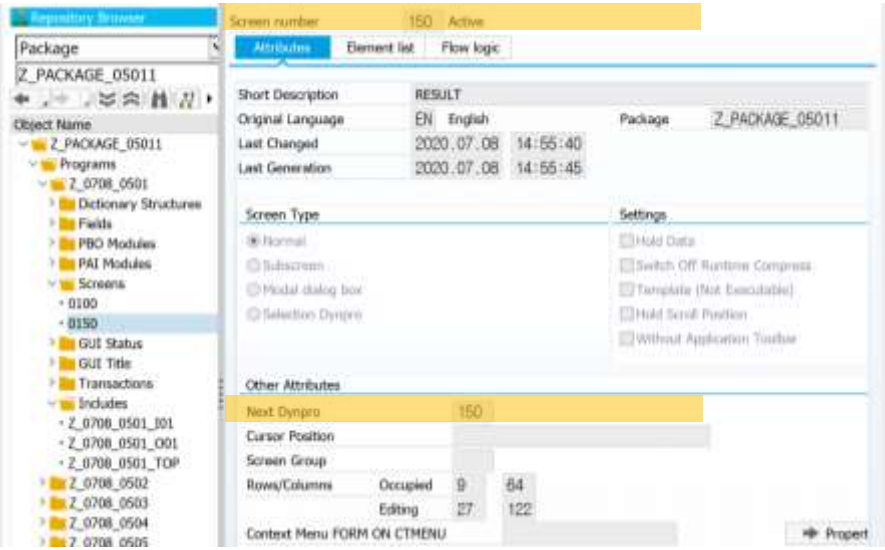
General | Texts/I/O Templates | Special | Display | Mod. Groups/Functions

H.. N Name	Type...	Line	C...	D...	Vi...	H...	Sc...	Format	In...	O...	Out...
SCARR-CARRID	Text	3	7	11	11	1					
SCARR-CARRID	I/O	3	24	3	3	1		CHAR	✓	✓	
SEARCH	Push	3	38	6	6	1					
OK_CODE	OK	0	0	20	20	1		OK			

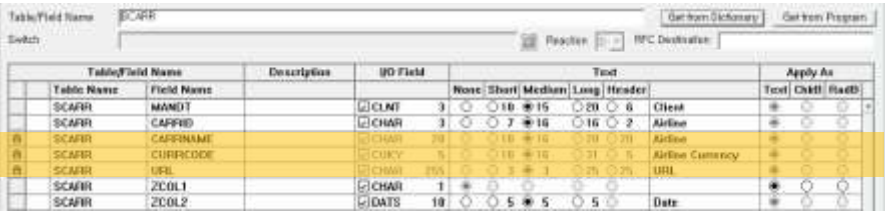
LAYOUT | FLOW LOGIC

```
PROCESS BEFORE OUTPUT.
MODULE STATUS_0100.

PROCESS AFTER INPUT.

MODULE USER_COMMAND_0100.
```

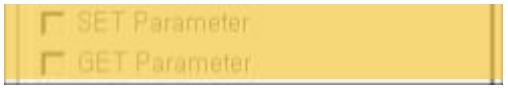
SCREEN | 150



LAYOUT |



OUTPUT |



LAYOUT | FLOW LOGIC

```
PROCESS BEFORE OUTPUT.
MODULE STATUS_0150.
MODULE RESULT_0150.
*
PROCESS AFTER INPUT.
MODULE USER_COMMAND_0150.
```

STATUS |

Z_0708_0501_TOP |

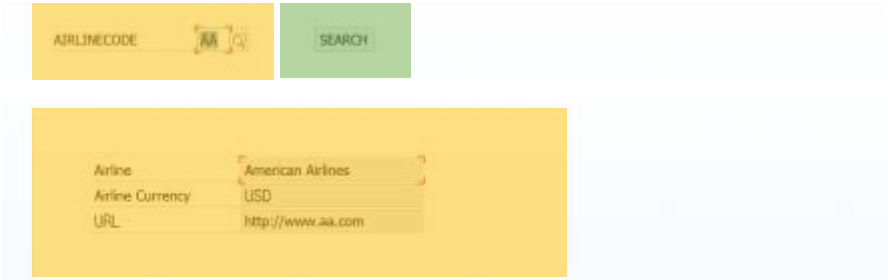
```
TABLES SCARR.
DATA OK_CODE TYPE SY-UCOMM.
DATA WA_SCARR TYPE SCARR.
```

Z_0708_0501_O01 |

```
MODULE STATUS_0100 OUTPUT.
            SET PF-STATUS '0100'.  "STATUS BACK 지정
            SET TITLEBAR '0100' WITH TEXT-VIE .  "TITILEBAR 지정
ENDMODULE.
*&---------------------------------------------------------------

MODULE STATUS_0150 OUTPUT.
            SET PF-STATUS '0150'.  "STATUS BACK 지정
            SET TITLEBAR '0150' WITH TEXT-VIE .  "TITILEBAR 지정
ENDMODULE.

*&---------------------------------------------------------------
MODULE RESULT_0150 OUTPUT.
           SELECT SINGLE *
                    FROM SCARR
                    INTO SCARR
                    WHERE CARRID = SCARR-CARRID.

*            CHECK SY-SUBRC <> 0 .
                IF SY-SUBRC <> 0.
                    CLEAR SCARR.
                    MESSAGE I007(ZBC4100) .
                ENDIF.
*                    CLEAR OK_CODE.
ENDMODULE.
```

Z_0708_0501_I01 |

```
MODULE USER_COMMAND_0100 INPUT.

  CASE OK_CODE.
    WHEN 'BACK'.  "BACK 버튼을 누를 시
      LEAVE TO SCREEN 0.  "호출 화면으로 돌아 감
    WHEN 'DISP'.  "SEARCH 버튼을 누를 시
      CALL SCREEN 150.  "150 SCREEN을 불러옴
    WHEN OTHERS.
  ENDCASE.

ENDMODULE.

*---------------------------------------------------------------
MODULE USER_COMMAND_0150 INPUT.
            IF OK_CODE = 'BACK'.  "BACK 버튼을 누를 시
                LEAVE TO SCREEN 0.  "프로그램을 종료
            ENDIF.
ENDMODULE.
```



Z_PACKAGE_05011 | Z_0708_0502 | 요구사항

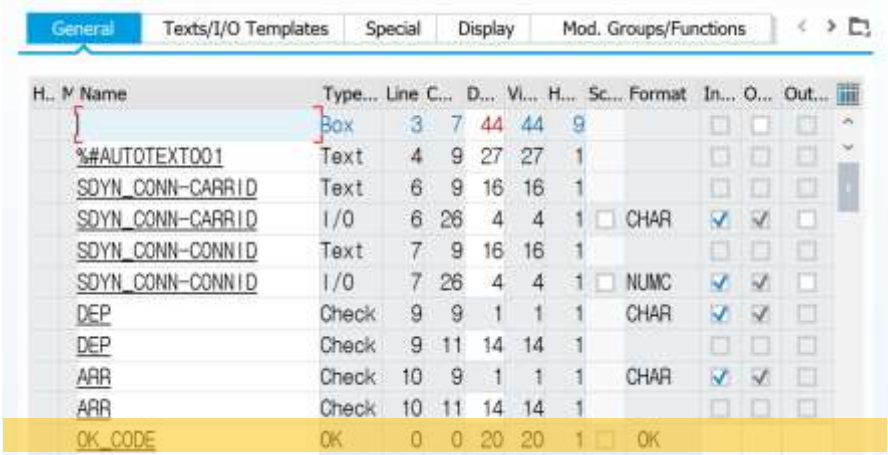
운항 일정 테이블 (SDYN_CONN) 에서 운항 일정이 존재하는 항공편에 대한 AIRLINE | FLIGHT CONNECTION NUMBER 를 입력했을 때 항공편 테이블 (SFLIGHT) 에서 조회한 AIRLINE | FLIGHT NUMBER 과 DEPARTURE 또는 ARRIVAL 정보를 조회하고자 한다. 화면은 사용자가 항공사 코드를 입력 받는 화면과 조회 화면으로 구성되어 있으며 상단 STATUS BAR의 BACK 버튼을 누르면 이전 화면으로 돌아가도록 구현한다. 입력 받는 화면은 INPUT 필드와 DEP | ARR 체크박스로 이루어져 있으며 해당 항목에 체크하면 조회 화면에서 DEP | DEPARTURE 또는 ARR | ARRIVAL 항목을 조회할 수 있다. DEPARTURE에서 조회되는 항목은 DEPARTURE | COUNTRY | DEPART.CITY | DEP.AIRPORT 이며 ARRIVAL 에서 조회되는 항목은 COUNTRY | ARRIVAL CITY | DEST. AIRPORT | ARRIVAL TIME으로 구성되어 있다.

TABLE & STR	SDYN | SFLIGHT
SCREEN	300 400
INPUT	AIRLINE | FLIGHT CONNECTION NUMBER
OUTPUT	AIRLINE | FLIGHT NUMBER
DEPARTURE	DEPARTURE | COUNTRY | DEPART. CITY | DEP.AIRPORT
ARRIVAL	COUNTRY | ARRIVAL CITY | DEST. AIRPORT | ARRIVAL TIME
STATUS	BACK
CHECK BOX	DEP | ARR

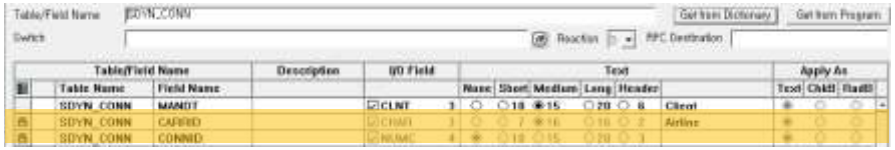
SE80 | MOD. POOL

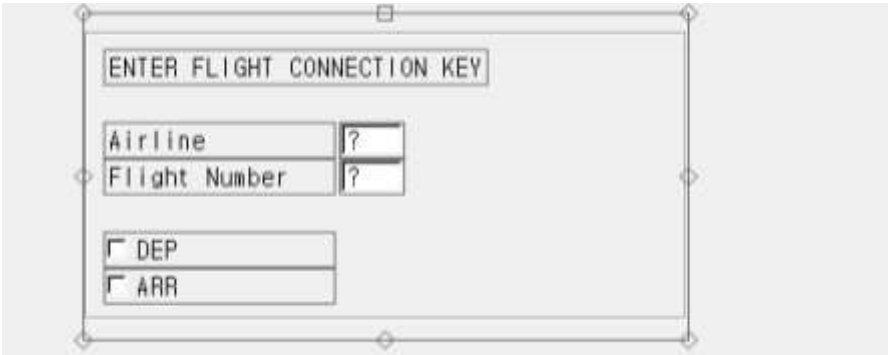
```
INCLUDE Z_0708_0502_TOP    .  " Global Data
INCLUDE Z_0708_0502_O01    .  " PBO-Modules
INCLUDE Z_0708_0502_I01    .  " PAI-Modules
```

SCREEN | 300 |

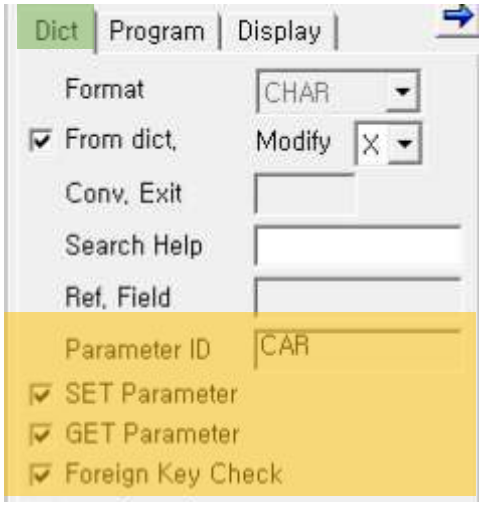


LAYOUT |

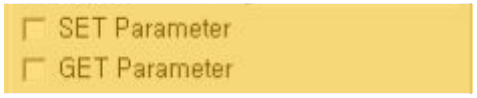




INPUT |



CHECKBOX |



LAYOUT | FLOW LOGIC

PROCESS BEFORE OUTPUT.

MODULE STATUS_0300. *"STATUS BACK 지정*

*

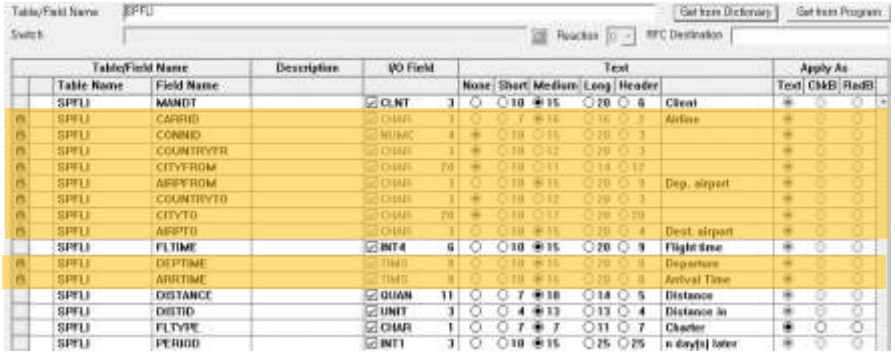
PROCESS AFTER INPUT.

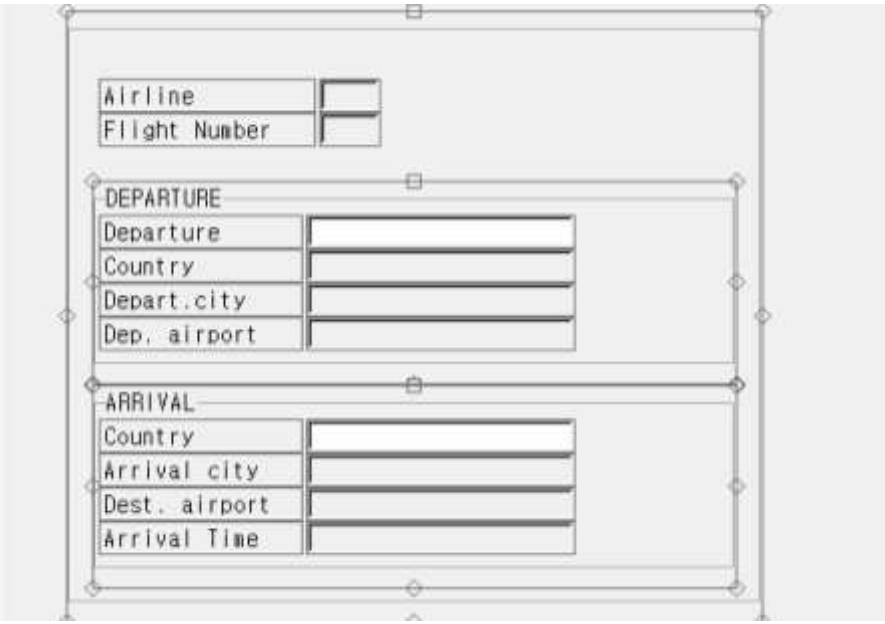
MODULE USER_COMMAND_0300.

SCREEN | 400 |

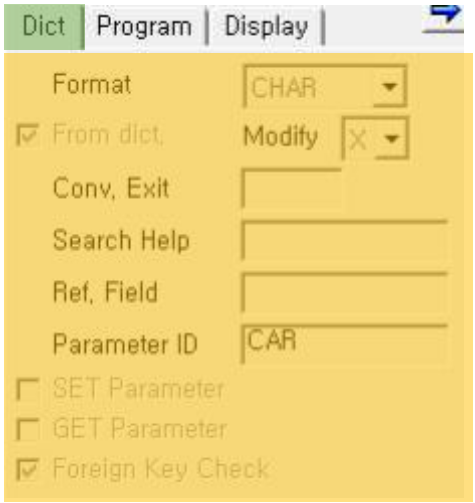
SPFLI-CITYTO	Text	16	8	15	15	1				
SPFLI-CITYTO	I/O	16	24	20	20	1		CHAR		✓
SPFLI-AIRPTO	Text	17	8	15	15	1				
SPFLI-AIRPTO	I/O	17	24	20	20	1		CHAR		✓
SPFLI-ARRTIME	Text	18	8	15	15	1				
SPFLI-ARRTIME	I/O	18	24	20	20	1		TIMS		✓
OK_CODE	OK	0	0	20	20	1		OK		

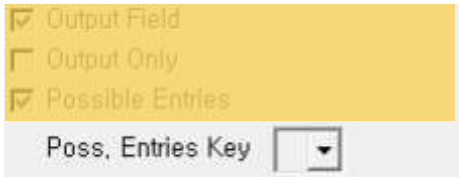
LAYOUT |

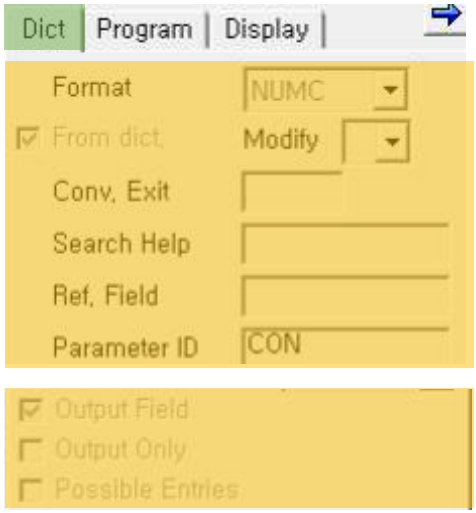




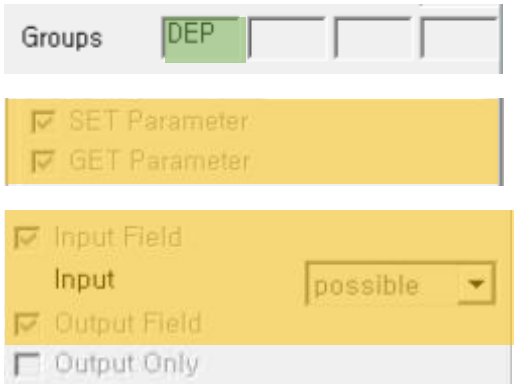
OUTPUT |



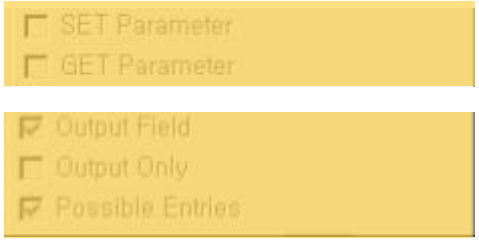




DEPARTURE | DEPARTURE

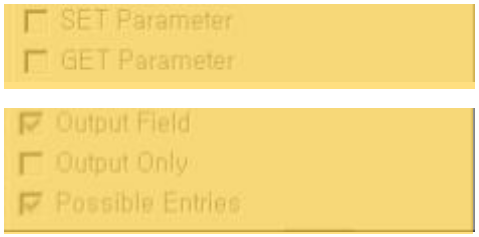


DEPARTURE |



ARRIVAL | ARR

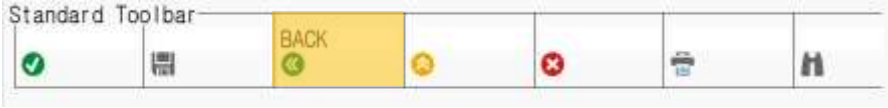
ARRIVAL |



LAYOUT | FLOW LOGIC

```
PROCESS BEFORE OUTPUT.
MODULE STATUS_0400.
MODULE GET_SPFLI_0400.
MODULE MODIFY_SCREEN.

PROCESS AFTER INPUT.
MODULE USER_COMMAND_0400.
```

STATUS |



Z_0708_0502_TOP |

```
TABLES: SPFLI.
TABLES: SDYN_CONN.
DATA WA_SPFLI TYPE SPFLI.
DATA OK_CODE TYPE SY-UCOMM.

DATA: DEP, ARR. "CHECKBOX
```

Z_0708_0502_O01 |

```
*&---------------------------------------------------------------------
MODULE STATUS_0300 OUTPUT.
 SET PF-STATUS '0300'.  "STATUS BACK 지정
ENDMODULE.
*&---------------------------------------------------------------------
MODULE STATUS_0400 OUTPUT.
  SET PF-STATUS '0400'.  "STATUS BACK 지정
            SELECT SINGLE *
                      FROM SPFLI
                      INTO SPFLI
                      WHERE CARRID = SDYN_CONN-CARRID
                           AND CONNID = SDYN_CONN-CONNID.
            IF SY-SUBRC <> 0.
                CLEAR SPFLI.
                MESSAGE I007(ZBC4100) .
            ENDIF.
ENDMODULE.
*&---------------------------------------------------------------------
MODULE MODIFY_SCREEN OUTPUT.
            LOOP AT SCREEN.
                    IF SCREEN-
GROUP1 = 'DEP' AND DEP IS INITIAL.  "CHECK X
                      SCREEN-ACTIVE = 0 .  "INVISIBLE
                    ELSEIF SCREEN-
GROUP1 = 'ARR' AND ARR IS INITIAL.
                        SCREEN-ACTIVE = 0 .  "INVISIBLE
                    ENDIF.
                    MODIFY SCREEN.
            ENDLOOP.
ENDMODULE.
```

Z_0708_0502_I01 |

```
*---------------------------------------------------------------------
MODULE USER_COMMAND_0300 INPUT.
            CALL SCREEN 400.  "400 스크린을 불러옴

            IF OK_CODE = 'BACK'.  "BACK 버튼을 누를 시
                LEAVE TO SCREEN 0.  "호출 화면으로 돌아 감
            ENDIF.
ENDMODULE.
*---------------------------------------------------------------------
MODULE USER_COMMAND_0400 INPUT.
            IF OK_CODE = 'BACK'.  "BACK 버튼을 누를 시
                LEAVE TO SCREEN 0.  "호출 화면으로 돌아 감
            ENDIF.
ENDMODULE.
```

Z_PACKAGE_05011 | Z_0708_0503 | 요구사항

SCREEN 100과 SCREEN 200으로 이루어진 프로그램을 구현하자. SCREEN 100 은 사용자가 입력할 수 있는 필드와 NEXT | MODI | DISP 버튼으로 이루어져 있으며 SCREEN 200 화면은 조회 및 수정을 할 수 있는 필드와 SAVE 버튼으로 이루어져 있다. 사용자가 SCREEN 100 에서 필드를 입력하고 NEXT 버튼을 누르면 필드에 입력한 값이 전달되며, SCREEN 200 의 필드는 활성화되지 않는다. 만약, 사용자가 MODI 버튼을 누르면 SCREEN 200의 필드는 활성화된다. DISP 버튼의 경우 필드는 비활성화되며, SCREEN 200에서 SAVE를 누르는 경우 필드에 입력된 값이 객체에 저장되고 상단 STATUS BAR의 BACK 버튼을 누르면 이전 화면으로 돌아가도록 구현한다.

SCREEN	100 200
INPUT	FIELD1
OUTPUT	FIELD2
BUTTON	NEXT | MODI | DISP
BUTTON2	SAVE
STATUS	BACK

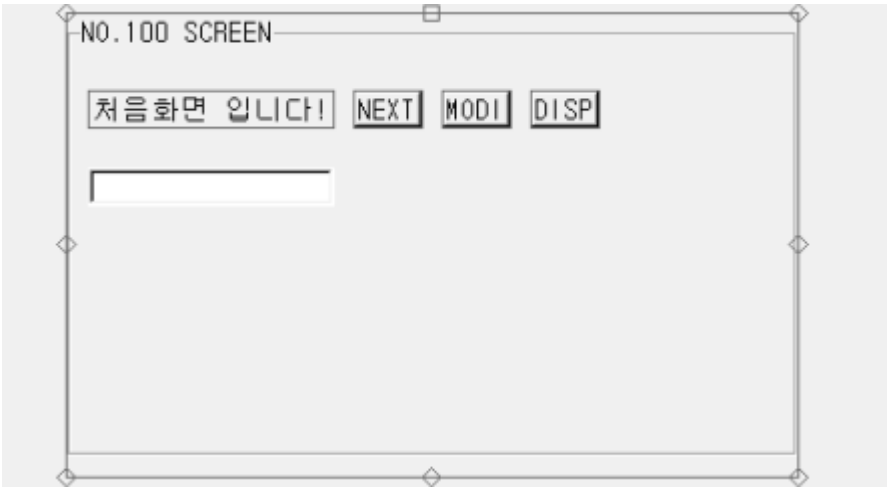
SE80 | MOD. POOL

```
INCLUDE Z_0708_0503_TOP      .   " Global Data
INCLUDE Z_0708_0503_O01      .   " PBO-Modules
INCLUDE Z_0708_0503_I01      .   " PAI-Modules
```

SCREEN | 100 | LAYOUT

FIELD1	I/O	6	8	16	16	1	☐	CHAR	☑	☑	☐
OK_CODE	OK	0	0	20	20	1	☐	OK			

LAYOUT |



INPUT |

Name FIELD1

SET Parameter
GET Parameter
Foreign Key Check

Input Field
Input possible
Output Field

BUTTON |

Name NEXT
Text NEXT

Switch Reaction D
FctCode NEXT FctType

SET Parameter
GET Parameter

Input Field
Input
Output Field
Output Only

Name MODI
Text MODI

FctCode MODI FctType

Name DISP
Text DISP

FctCode DISP FctType

LAYOUT | FLOW LOGIC

PROCESS BEFORE OUTPUT.
MODULE STATUS_0100.
*
PROCESS AFTER INPUT.
MODULE USER_COMMAND_0100.

SCREEN | 200 | LAYOUT

SAVE	Push	6	25	4	4	1				
OK_CODE2	OK	0	0	20	20	1		OK		

LAYOUT |

OUTPUT |

Name FIELD2

☐ SET Parameter
☐ GET Parameter

☐ Input Field
Input not possibl
☑ Output Field
☐ Output Only

BUTTON2 |

Name SAVE
Text SAVE

FctCode SAVE FctType

☐ SET Parameter
☐ GET Parameter

☐ Input Field
Input
☐ Output Field
☐ Output Only

LAYOUT | FLOW LOGIC

PROCESS BEFORE OUTPUT.
MODULE STATUS_0200.
MODULE MODIFY_SCREEN.
*
PROCESS AFTER INPUT.
MODULE USER_COMMAND_0200.

STATUS |

Z_0708_0503_TOP |

```
PROGRAM Z_0708_0503.

DATA OK_CODE TYPE SY-UCOMM.
DATA OK_CODE2 TYPE SY-UCOMM.
DATA: FIELD1(20). "INPUT FIELD 지정
```

Z_0708_0503_O01 |

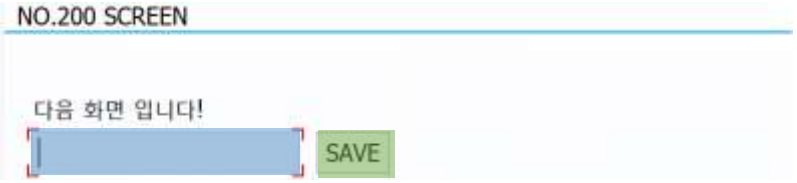
```
*&---------------------------------------------------------------------
MODULE STATUS_0100 OUTPUT.
SET PF-STATUS '0100'. "STATUS BACK 지정
ENDMODULE.
*&---------------------------------------------------------------------
MODULE STATUS_0200 OUTPUT.
SET PF-STATUS '0200'. "STATUS BACK 지정
ENDMODULE.
*&---------------------------------------------------------------------
MODULE MODIFY_SCREEN OUTPUT. "수정버튼 입력 시 조건
        LOOP AT SCREEN.
                IF OK_CODE = 'MODI' AND SCREEN-
NAME = 'FIELD2'.
                    SCREEN-INPUT = 1.
                ENDIF.
                MODIFY SCREEN.
*               CLEAR OK_CODE2.
        ENDLOOP.
ENDMODULE.
```

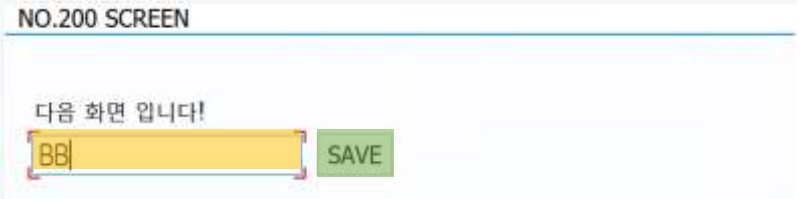
Z_0708_0503_I01 |

```
MODULE USER_COMMAND_0100 INPUT.
        CASE OK_CODE.
            WHEN 'BACK'.
                LEAVE PROGRAM. "프로그램을 종료
            WHEN 'NEXT'.
                CALL SCREEN 200. "200 스크린을 불러옴
            WHEN 'MODI' OR 'DISP'.
                CALL SCREEN 200. "200 스크린을 불러옴
        ENDCASE.
ENDMODULE.
*&---------------------------------------------------------------------

MODULE USER_COMMAND_0200 INPUT.
        CASE OK_CODE2.
            WHEN 'BACK'.
                LEAVE TO SCREEN 0 . "호출 시점으로 감
            WHEN 'SAVE'.
                CLEAR OK_CODE2.
```

```
            ENDCASE.
ENDMODULE.

*-----------------------------------------------------------
MODULE INPUT_CHECK INPUT. "INPUT 필드에 입력 확인
            MESSAGE 'ERROR' TYPE 'E'.
ENDMODULE.
```

St...	Variable	V...	Val.
	FIELD1		AA





St...	Variable	V...	Val.
	FIELD1		BB

St...	Variable	V...	Val.
	FIELD1		BB

Z_PACKAGE_05011 | Z_0708_0505 | 요구사항

필드가 4개가 있는 화면의 필드의 INPUT 값 체크를 할 수 있도록 로직을 구현해 보자. 각각의 필드는 DEFAULT 값으로 A B C D 의 값을 가지며, 필드 1은 값 | A를 입력하지 않으면 다른 필드가 비 활성화되며 경고 메시지를 출력한다. 필드 2는 값 | B를 입력하지 않으면 다른 필드가 비 활성화되며 경고 메시지를 출력한다. 필드 3 과 필드 2는 각각 값 C | B를 입력하지 않거나 둘 중 하나의 값만 입력되지 않아도 다른 필드가 비 활성화되며 경고 메시지를 출력한다. 이와 같이 필드 3과 필드 4는 각각 값 C | D를 입력하지 않거나 둘 중 하나의 값만 입력되지 않아도 다른 필드가 비 활성화되며 경고 메시지를 출력한다. 상단 STATUS BAR 의 BACK | CANC 버튼을 누르면 이전 화면으로, EXIT 버튼을 누르면 호출한 시점으로 돌아가도록 구현한다.

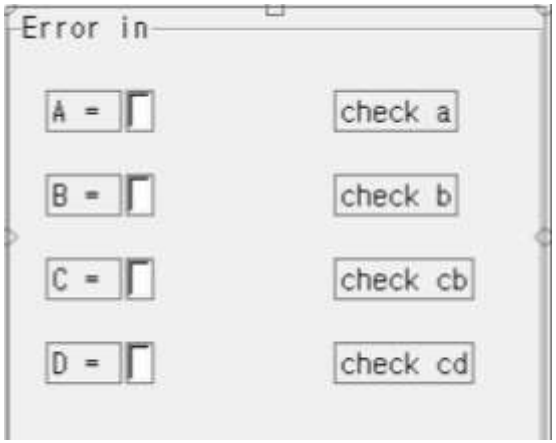
SCREEN	100
INPUT | OUTPUT	FIELD1 | FIELD2 | FIELD3 | FIELD4 A | B | C | D
STATUS	BACK EXIT CANC

SE80 | MOD. POOL

```
INCLUDE Z_0708_0505_TOP.
INCLUDE Z_0708_0505_O01.
INCLUDE Z_0708_0505_I01.
```

SCREEN | 100 |

LAYOUT |



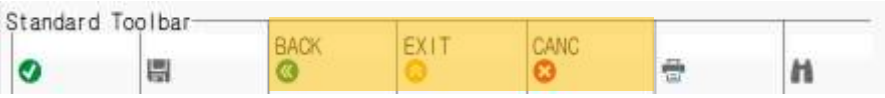
INPUT & OUTPUT |

LAYOUT | FLOW LOGIC

```
PROCESS BEFORE OUTPUT.
MODULE SET_STATUS_100.
MODULE INIT.
*
PROCESS AFTER INPUT.
MODULE EXIT.
```

```
FIELD A MODULE CHECK_A.
FIELD B MODULE CHECK_B.

CHAIN.
  FIELD: C, B
  MODULE CHECK_CB.
ENDCHAIN.

CHAIN.
  FIELD: C, D
  MODULE CHECK_CD.
ENDCHAIN.

MODULE SUCCESS.
```

STATUS |



Z_0708_0505_TOP |

```
PROGRAM Z_0708_0505.
DATA: OK_CODE LIKE SY-UCOMM.
DATA: A, B.
DATA: C, D.
```

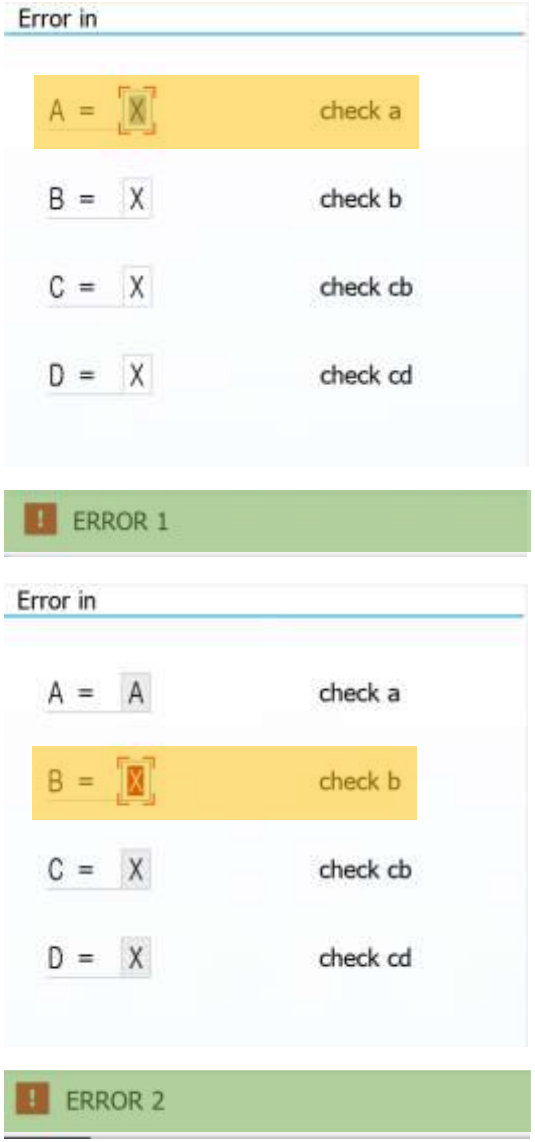
Z_0708_0505_O01 |

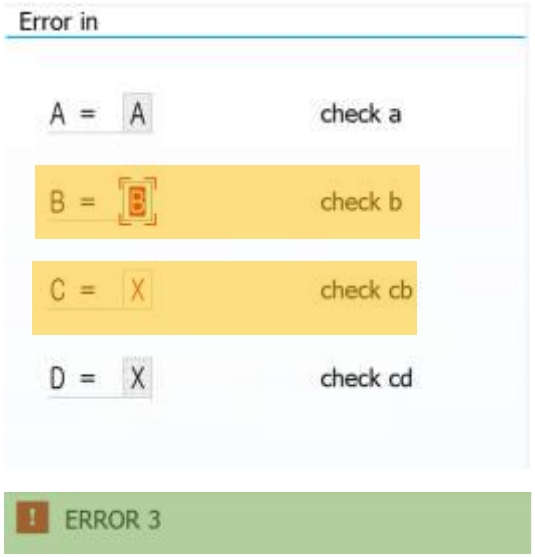
```
*---------------------------------------------------------------
MODULE INIT OUTPUT.
  A = 'X'.
  B = 'X'.
  C = 'X'.
  D = 'X'.
ENDMODULE.                          " INIT  OUTPUT
*---------------------------------------------------------------
MODULE SET_STATUS_100 OUTPUT.
  SET PF-STATUS 'DYNPRO'.
  SET TITLEBAR 'RETURN'.
ENDMODULE.                          " SET_STATUS_100  OUTPUT
```

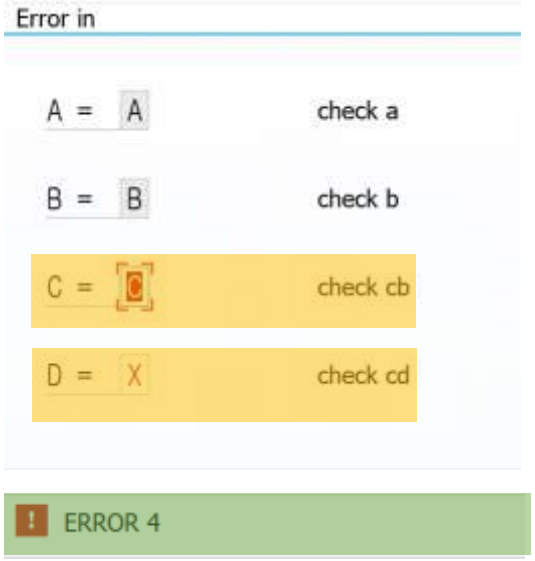
Z_0708_0505_I01 |

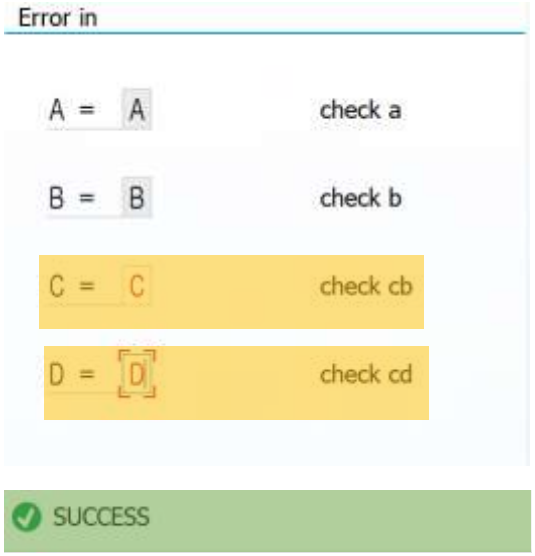
```
*---------------------------------------------------------------
MODULE EXIT INPUT.
  IF OK_CODE = 'EXIT'.
    LEAVE PROGRAM.
  ENDIF.
ENDMODULE.                          " EXIT  INPUT
*---------------------------------------------------------------
```

```
MODULE SUCCESS INPUT.
  MESSAGE 'SUCCESS' TYPE 'S'.
  LEAVE PROGRAM.
ENDMODULE.                          " USER_COMMAND_0200  INPUT
*&---------------------------------------------------------------------
MODULE CHECK_A INPUT.
  CHECK A <> 'A'.
  MESSAGE 'ERROR 1' TYPE 'E'.
ENDMODULE.                          " CHECK_A  INPUT
*&---------------------------------------------------------------------
MODULE CHECK_CB INPUT.
  CHECK C NE 'C'.
  MESSAGE 'ERROR 3' TYPE 'E'.
ENDMODULE.                          " CHECK_CB  INPUT
*&---------------------------------------------------------------------
MODULE CHECK_CD INPUT.
  CHECK D NE 'D'.
  MESSAGE 'ERROR 4' TYPE 'E'.
ENDMODULE.                          " CHECK_CD  INPUT
```









Z_PACKAGE_05011 | Z_0708_0506 | 요구사항

덧셈을 할 수 있는 화면을 구현해보자. 사용자는 2개의 INPUT 필드에 원하는 값을 입력할 수 있고 두개의 INPUT FIELD에 입력된 값은 더하기를 수행해 아래의 OUTPUT 필드에 표출된다. 상단 STATUS BAR의 BACK | CANC 버튼을 누르면 이전 화면으로, EXIT 버튼을 누르면 호출한 시점으로 돌아가도록 구현한다.

SCREEN	100
INPUT	FIELD1 | FIELD2
OUTPUT	FIELD3
STATUS	BACK EXIT CANC

SE80 | MOD. POOL

```
INCLUDE Z_0708_0506_TOP      . " Global Data
INCLUDE Z_0708_0506_O01      . " PBO-Modules
INCLUDE Z_0708_0506_I01      . " PAI-Modules
```

SCREEN | 100 |

OK_CODE		OK	0	0	20	20	1	☐	OK

LAYOUT |

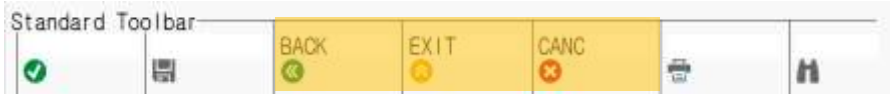
INPUT |

OUTPUT |

LAYOUT | FLOW LOGIC

```
PROCESS BEFORE OUTPUT.
MODULE STATUS_0100.

PROCESS AFTER INPUT.
MODULE EXIT AT EXIT-COMMAND.
CHAIN.
    FIELD: FIELD1, FIELD2 MODULE INPUT_SUM . "REQUEST
ENDCHAIN.
MODULE USER_COMMAND_0100.
```

STATUS |



Z_0708_0506_TOP |

```
PROGRAM Z_0708_0506.
DATA OK_CODE TYPE SY-UCOMM.
```

```
DATA IO_COMMAND.
DATA: FIELD1 TYPE I,
      FIELD2 TYPE I,
      FIELD3 TYPE I.
```

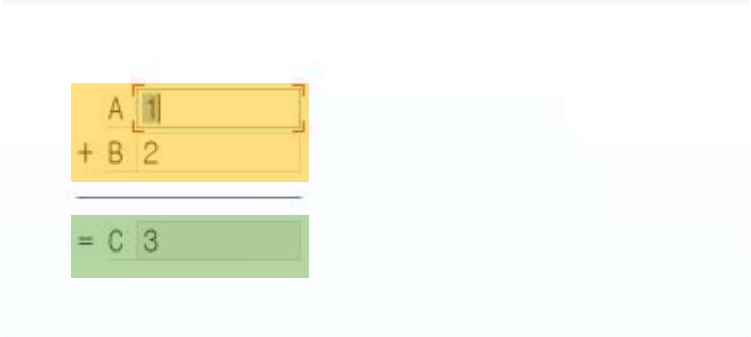
Z_0708_0506_O01 |

```
MODULE STATUS_0100 OUTPUT.
SET PF-STATUS '0100'.
* SET TITLEBAR 'xxx'.
ENDMODULE.
```

Z_0708_0506_I01 |

```
*-----------------------------------------------------------
MODULE USER_COMMAND_0100 INPUT.
      DATA LV_ANSWER.
      CASE OK_CODE.
        WHEN 'BACK'.
        CALL FUNCTION 'POPUP_TO_CONFIRM'
        EXPORTING
            TEXT_QUESTION          = 'SAVE DATA?'
            TEXT_BUTTON_1          = 'YES'(001)
              TEXT_BUTTON_2          = 'NO'(002)
              DEFAULT_BUTTON          = '1'
              DISPLAY_CANCEL_BUTTON      = 'X'
          IMPORTING
              ANSWER = LV_ANSWER
          EXCEPTIONS
              TEXT_NOT_FOUND = 1
          OTHERS = 2  .
          CHECK LV_ANSWER = '1'.
          LEAVE PROGRAM.
      ENDCASE.
ENDMODULE.
*-----------------------------------------------------------
MODULE EXIT INPUT.
      CASE OK_CODE.
          WHEN 'EXIT'.
            LEAVE TO SCREEN 0.
          WHEN 'CANC'.
            LEAVE PROGRAM.
      ENDCASE.
ENDMODULE.
*-----------------------------------------------------------
MODULE INPUT_SUM INPUT.
      FIELD3 = FIELD1 + FIELD2.
ENDMODULE.
```

A
+ B
= C 0


A 1
+ B 2
= C 3

SAP
SAVE DATA?
YES
NO
Cancel