

SDYN_CONN	FLDATE	DATE	10	0	0	5	9	0	18	0	0	Date	*	0	0	0
SDYN_CONN	PRICE	CURR	20	0	0	5	9	0	9	0	0	Airfare	*	0	0	0
SDYN_CONN	CURRENCY	CURCY	5	0	0	18	18	0	31	0	0	Airline Currency	*	0	0	0
SDYN_CONN	PLANETYPE	CHAP	10	0	0	7	11	0	11	0	0	Plane Type	*	0	0	0
SDYN_CONN	SEATSMAX	CHT4	10	0	0	8	20	0	31	0	0	Max. capacity econ.	*	0	0	0
SDYN_CONN	SEATSOCC	CHT4	10	0	0	18	18	0	30	0	0	Occupied econ.	*	0	0	0
SDYN_CONN	PAYMENTSUM	CURR	22	0	0	5	10	0	18	0	13	Total	*	0	0	0
SDYN_CONN	CARRNAME	CHAR	20	0	0	18	18	0	20	0	0	Airline	*	0	0	0
SDYN_CONN	MARK	CHAR	1	0	0	18	15	0	20	0	0	Selection	*	0	0	0

LAYOUT |

Flow logic Attributes Element List

Name: SDYN_CONN-CARRID Text: Lg 3 H 1 L 4 C 20

Airline	?
Flight Number	?
Date	?
Airfare	
Airline Currency	
Plane Type	
Max. capacity econ.	
Occupied econ.	
Total	

COMMAND |

Elem. Type Input/Output Field

Name: JO_COMMAND

Attributes

Dict Program Display

☒ Input Field

Input: possible

☒ Output Field

INPUT |

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 |

☐ SET Parameter

☐ GET Parameter

☐ Input Field

Input: not possible

☒ Output Field

☐ Output Only

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.

Airline	AA
Flight Number	0017
Date	2020.01.09
Airfare	0.00
Airline Currency	
Plane Type	
Max. capacity econ.	0
Occupied econ.	0
Total	0.00

Airline	AA
Flight Number	17
Date	2020.01.09
Airfare	422.94
Airline Currency	USD
Plane Type	747-400
Max. capacity econ.	385
Occupied econ.	385
Total	199,357.26

X 입력 시 종료

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

INCLUDE Z_0708_0501_O01

INCLUDE Z_0708_0501_I01

. " Global Data

. " PBO-Modules

. " PAI-Modules

SCREEN | 100

Repository Browser: Screen number 100 Active

Package: Z_PACKAGE_05011

Object Name: Z_PACKAGE_05011

Attributes:

- Short Description: Z_0708_0501
- Original Language: EN English
- Last Changed: 2020.07.08 15:03:56
- Last Generation: 2020.07.08 15:03:57

Other Attributes:

- Next Dynpro: 100
- Cursor Position: 1
- Screen Group: 1
- Rows/Columns: Occupied 3 43, Editing 27 122
- Context Menu: FORM ON CTMENU

LAYOUT |

Table/Field Name: SCARR

Table Name	Field Name	Description	UD Field	None	Short	Medium	Long	Header	Apply As
SCARR	MANDT		CLNT	3	10	15	20	6	Client
SCARR	CARRID		CHAR	3	10	15	20	6	Airline
SCARR	CARRNAME		CHAR	20	10	15	20	20	Airline Currency
SCARR	CURRCODE		CURCY	5	10	15	20	5	URL
SCARR	URL		CHAR	255	3	3	25	25	Date
SCARR	ZCOL1		CHAR	1	10	15	20	5	
SCARR	ZCOL2		DATS	10	5	5	5	5	

Name: SCARR-CARRID Text: Lg 3 H 1 L 3 Cl 24

Layout Preview: AIRLINE CODE ? SEARCH

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...
		<u>SCARR-CARRID</u>	<u>Text</u>	3	7	11	11	1					
		<u>SCARR-CARRID</u>	<u>I/O</u>	3	24	3	3	1	<input type="checkbox"/>	CHAR	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		<u>SEARCH</u>	<u>Push</u>	3	38	6	6	1			<input type="checkbox"/>	<input type="checkbox"/>	
		<u>OK_CODE</u>	<u>OK</u>	0	0	20	20	1	<input type="checkbox"/>	OK			

LAYOUT | FLOW LOGIC

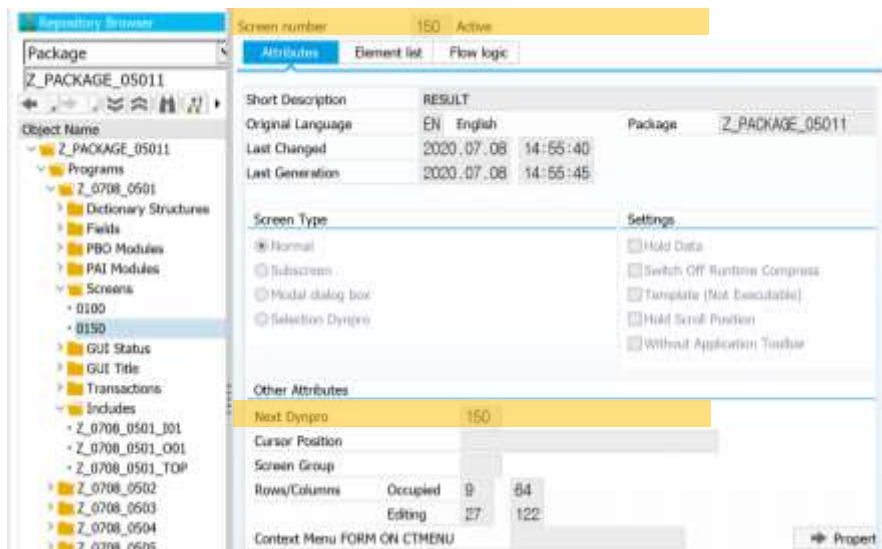
PROCESS BEFORE **OUTPUT**.

MODULE STATUS_0100.

PROCESS AFTER **INPUT**.

MODULE USER_COMMAND_0100.

SCREEN | 150

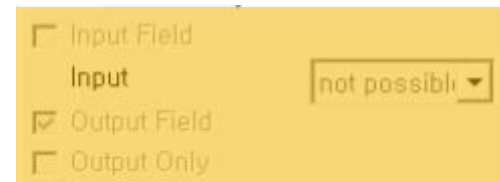
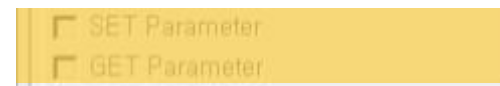


LAYOUT |

Table/Field Name	Field Name	Description	ID Field	None	Short	Medium	Long	Header	Apply As
SCARR	MANDT		CLNT	3	10	15	20	6	Client
SCARR	CAFRID		CHAR	3	7	14	16	2	Airline
SCARR	CAFRNAME		CHAR	20	10	14	20	20	Airline
SCARR	CURRCD		CHAR	5	10	14	21	5	Airline Currency
SCARR	URL		CHAR	255	3	3	25	25	URL
SCARR	ZDOL1		CHAR	1	5	5	5	5	Date
SCARR	ZDOL2		DATS	10	5	5	5	5	Date



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 . "TITLEBAR 지정

ENDMODULE.

*&-----

MODULE STATUS_0150 OUTPUT.

SET PF-STATUS '0150'. "STATUS BACK 지정

SET TITLEBAR '0150' WITH TEXT-VIE . "TITLEBAR 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 |

H...	N Name	Type...	Line	C...	D...	V...	H...	Sc...	Format	In...	O...	Out...
	%#AUTOTEXT001	Text	4	9	27	27	1					
	SDYN_CONN-CARRID	Text	6	9	16	16	1					
	SDYN_CONN-CARRID	I/O	6	26	4	4	1		CHAR	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	SDYN_CONN-CONNID	Text	7	9	16	16	1					
	SDYN_CONN-CONNID	I/O	7	26	4	4	1		NUMC	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	DEP	Check	9	9	1	1	1		CHAR	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	DEP	Check	9	11	14	14	1					
	ARR	Check	10	9	1	1	1		CHAR	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	ARR	Check	10	11	14	14	1					
	OK_CODE	OK	0	0	20	20	1		OK			

LAYOUT |

ENTER FLIGHT CONNECTION KEY

Airline ?

Flight Number ?

☐ DEP

☐ ARR

INPUT |

Dict Program Display

Format CHAR

☒ From dict, Modify X

Conv. Exit

Search Help

Ref. Field

Parameter ID CAR

☒ SET Parameter

☒ GET Parameter

☒ Foreign Key Check

Dict Program Display

☒ Input Field

Input required

☒ Output Field

☐ Output Only

☒ Possible Entries

CHECKBOX |

☐ SET Parameter

☐ GET Parameter

☒ Input Field

Input

☒ Output Field

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	<input type="checkbox"/> CHAR	<input checked="" type="checkbox"/>	
SPFLI-AIRPTO	Text	17	8	15	15	1			
SPFLI-AIRPTO	I/O	17	24	20	20	1	<input type="checkbox"/> CHAR	<input checked="" type="checkbox"/>	
SPFLI-ARRTIME	Text	18	8	15	15	1			
SPFLI-ARRTIME	I/O	18	24	20	20	1	<input type="checkbox"/> TMS	<input checked="" type="checkbox"/>	
OK_CODE	OK	0	0	20	20	1	<input type="checkbox"/> OK		

LAYOUT |

Table/Field Name: SPFLI

Switch: ☐ Reached ☐ RFC Destination

Table Name	Field Name	Description	WO Field	None	Short	Medium	Long	Header	Text	Apply As
SPFLI	MANDT		<input checked="" type="checkbox"/> CLNT	3	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	70	8
SPFLI	CARRID		<input checked="" type="checkbox"/> CARR	3	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	16	3
SPFLI	CARRID		<input checked="" type="checkbox"/> NUMC	4	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	20	3
SPFLI	COUNTRYR		<input checked="" type="checkbox"/> CARR	3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	20	3
SPFLI	CITYFROM		<input checked="" type="checkbox"/> CARR	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	14	12
SPFLI	APPTFROM		<input checked="" type="checkbox"/> CARR	3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	20	8
SPFLI	COUNTRYTO		<input checked="" type="checkbox"/> CARR	3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	20	3
SPFLI	CITYTO		<input checked="" type="checkbox"/> CARR	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	20	20
SPFLI	ARRPTO		<input checked="" type="checkbox"/> CARR	3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	20	4
SPFLI	FLTIME		<input checked="" type="checkbox"/> INT4	6	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	20	8
SPFLI	DEPTIME		<input checked="" type="checkbox"/> TMS	8	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	20	8
SPFLI	ARRTIME		<input checked="" type="checkbox"/> TMS	8	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	20	8
SPFLI	DISTANCE		<input checked="" type="checkbox"/> QUAN	11	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	14	5
SPFLI	DISTID		<input checked="" type="checkbox"/> UNIT	3	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	13	4
SPFLI	FLTYPE		<input checked="" type="checkbox"/> CHAR	1	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	11	7
SPFLI	PERIOD		<input checked="" type="checkbox"/> INT1	3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	25	25

Airline	
Flight Number	
DEPARTURE	
Departure	
Country	
Depart. city	
Dep. airport	
ARRIVAL	
Country	
Arrival city	
Dest. airport	
Arrival Time	

OUTPUT |

Dict	Program	Display
Format <input type="text" value="CHAR"/>		
<input checked="" type="checkbox"/> From dict.	Modify	<input type="text" value="X"/>
Conv. Exit <input type="text"/>		
Search Help <input type="text"/>		
Ref. Field <input type="text"/>		
Parameter ID <input type="text" value="CAR"/>		
<input type="checkbox"/> SET Parameter		
<input type="checkbox"/> GET Parameter		
<input checked="" type="checkbox"/> Foreign Key Check		

<input checked="" type="checkbox"/> Output Field
<input type="checkbox"/> Output Only
<input checked="" type="checkbox"/> Possible Entries
Poss. Entries Key <input type="text"/>

Dict	Program	Display
------	---------	---------

Format	<input type="text" value="NUMC"/>
<input checked="" type="checkbox"/> From dict.	Modify <input type="text"/>
Conv. Exit <input type="text"/>	
Search Help <input type="text"/>	
Ref. Field <input type="text"/>	
Parameter ID <input type="text" value="CON"/>	

<input checked="" type="checkbox"/> Output Field
<input type="checkbox"/> Output Only
<input type="checkbox"/> Possible Entries

DEPARTURE | DEPARTURE

Groups	DEP		
--------	-----	--	--

<input checked="" type="checkbox"/> SET Parameter
<input checked="" type="checkbox"/> GET Parameter

<input checked="" type="checkbox"/> Input Field
Input <input type="text" value="possible"/>
<input checked="" type="checkbox"/> Output Field
<input type="checkbox"/> Output Only

DEPARTURE |

☐ SET Parameter
☐ GET Parameter

☒ Output Field
☐ Output Only
☒ Possible Entries

ARRIVAL | ARR

Groups **ARR**

☒ SET Parameter
☒ GET Parameter

☒ Input Field
Input
☒ Output Field

ARRIVAL |

☐ SET Parameter
☐ GET Parameter

☒ Output Field
☐ Output Only
☒ Possible Entries

LAYOUT | FLOW LOGIC

PROCESS BEFORE **OUTPUT**.

MODULE STATUS_0400.

MODULE GET_SPFLI_0400.

MODULE MODIFY_SCREEN.

PROCESS AFTER **INPUT**.

MODULE USER_COMMAND_0400.

STATUS |

Standard Toolbar

		BACK 				
--	--	-----------------	--	--	--	--

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.

ENTER FLIGHT CONNECTION KEY

Airline

AA

Flight Number

0017

☐ DEP
 ☐ ARR

Airline

AA

Flight Number

17

☒ DEP
 ☒ ARR

DEPARTURE

Departure

11:00:00

Country

US

Depart.city

NEW YORK

Dep. airport

JFK

ARRIVAL

Country

US

Arrival city

SAN FRANCISCO

Dest. airport

SFO

Arrival Time

14:01:00

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 |

NO.100 SCREEN

처음화면 입니다! NEXT MODI DISP

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
<input type="checkbox"/> SET Parameter	
<input type="checkbox"/> GET Parameter	
<input type="checkbox"/> Input Field	
Input	not possible
<input checked="" type="checkbox"/> Output Field	
<input type="checkbox"/> Output Only	

BUTTON2 |

Name	SAVE
Text	SAVE
FctCode	SAVE
FctType	
<input type="checkbox"/> SET Parameter	
<input type="checkbox"/> GET Parameter	
<input type="checkbox"/> Input Field	
Input	
<input type="checkbox"/> Output Field	
<input type="checkbox"/> 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.

NO.100 SCREEN

처음화면 입니다!

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

NO.200 SCREEN

다음 화면 입니다!

NO.100 SCREEN

처음화면 입니다!

NO.200 SCREEN

다음 화면 입니다!

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

NO.100 SCREEN

처음화면 입니다!

NO.200 SCREEN

다음 화면 입니다!

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 BAR의 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 |

OK_CODE	OK	0	0	20	20	1	OK
---------	----	---	---	----	----	---	----

LAYOUT |

INPUT & OUTPUT |

LAYOUT | FLOW LOGIC

PROCESS BEFORE OUTPUT.

MODULE SET_STATUS_100.

MODULE INIT.

*

PROCESS AFTER INPUT.

MODULE EXIT.

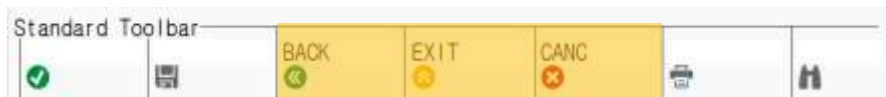
```
FIELD A MODULE CHECK_A.  
FIELD B MODULE CHECK_B.
```

```
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

```

Error in

A =	<input checked="" type="checkbox"/>	check a
B =	<input type="checkbox"/>	check b
C =	<input type="checkbox"/>	check cb
D =	<input type="checkbox"/>	check cd

! ERROR 1

Error in

A =	<input type="checkbox"/>	check a
B =	<input checked="" type="checkbox"/>	check b
C =	<input type="checkbox"/>	check cb
D =	<input type="checkbox"/>	check cd

! ERROR 2

Error in

A = A check a

B = B check b

C = X check cb

D = X check cd

! ERROR 3

Error in

A = A check a

B = B check b

C = B check cb

D = X check cd

! ERROR 4

Error in

A = A check a

B = B check b

C = C check cb

D = D check cd

✓ SUCCESS

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

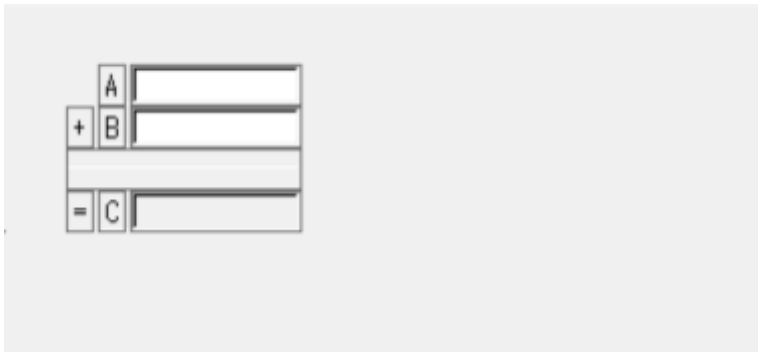
SE80 | MOD. POOL

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

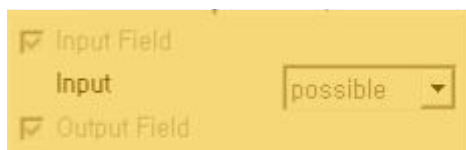
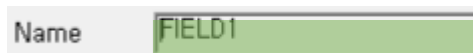
SCREEN | 100 |



LAYOUT |



INPUT |



OUTPUT |



LAYOUT | FLOW LOGIC

PROCESS BEFORE OUTPUT.

MODULE STATUS_0100.

PROCESS AFTER INPUT.

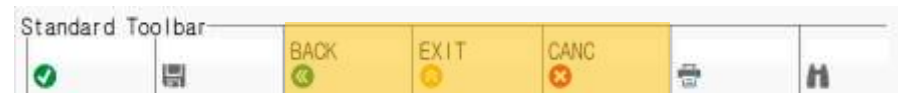
MODULE EXIT AT EXIT-COMMAND.

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.
```

$$\begin{array}{r} A \\ + B \\ \hline = C \end{array}$$

$$\begin{array}{r} A \\ + B \\ \hline = C \end{array}$$

 SAP ×

 SAVE DATA?

YES

NO

 Cancel