

Assignment on Module Pool with BDC

By Pavan Golesar

APPLICATION AREAS: SE80, SHDB, MB01, SE41, SE51, SE93.

SUMMARY:

This article demonstrates in a step-by-step process to first write BDC reports to do data transfer from legacy system, which is important work during SAP implementation projects and then have a module pool with help of dialog transaction ZMB01.

We call a user defined transaction code ZMB01 (INSTEAD OF MB01/MIGO) for ***bulk GRN***, Further we push the data to MB01 using Call transaction method of BDC.

Author(s): Pavan Golesar

Created on: 23February 2015



Table of Contents

Overview: How BDC Works.....

Steps in data transfer.....

Data Transfer Methods.....

Interview Questions (BDC).....

Module Pool Introduction.....

Flow of Program.....

Components in Module pool.....

Interview Questions (MP).....

Problems.....

Solutions.....



How BDC works:

BDC Reports are used to transfer data from legacy system into SAP system during SAP Implementation. Based on the functional requirements you have to now write a BDC report to upload the legacy data into SAP system.

To Perform Data transfer, we will perform 3 tasks:

Task 1: Do a recording for the transaction by supplying sample data.

Task 2: Create an ABAP report based on the recording done.

Task 3: Now write ABAP code to fetch data from legacy system and store into SAP System.



Steps in Data Transfer

- I. **Analysis** and cleanup of data in the non-SAP system
- II. **Extraction** of data from the non-SAP system
- III. **Mapping** the data in SAP format
- IV. **Transferring** the data to the SAP System
- V. **Checking** the data



Data Transfer Methods

Call transaction

Asynchronous processing takes place and the transaction is called every time.

Session method

Synchronous processing takes place and a session is prepared, which is to be handled through SM35.

Moreover, in Batch Input processing system log is maintained for every transaction. In Call transaction method messages have to be captured at runtime.

Direct input

This is an SAP supplied conversion program that allows very fast loading of certain objects. The disadvantage is that it is not customizable.



INTERVIEW QUESTIONS

1 What should be the approach for writing a BDC program?

Ans.: 1. Analysis the Data.2. Generate SAP structure.3. Develop transfer program
4. Create sequential file. 5. Create batch input program. 6. Process batch input
data

2. What are the steps in a BDC session ?

The first step in a BDC session is to identify the screens of the transaction that the program will process. Next step is to write a program to build the BDC table that will be used to submit the data to SAP. The final step is to submit the BDC table to the system in the batch mode or as a single transaction by the CALL TRANSACTION command.

3 What are the problems in processing batch input sessions? How is batch input process different from processing on line?

Ans.: Sessions cannot be run in parallel and not fast.

4 What do you do when the system crashes in the middle of a BDC batch session?

Check no. of records already updated and delete them from input file and run BDC again.



Module Pool Introduction:

Dialog programs are created with type as 'M' - Module Pool. They cannot be executed independently and must be attached to at least one transaction code in which you specify an initial screen.

SAP-ABAP supports two types of programs - Report Program and Dialog Program.

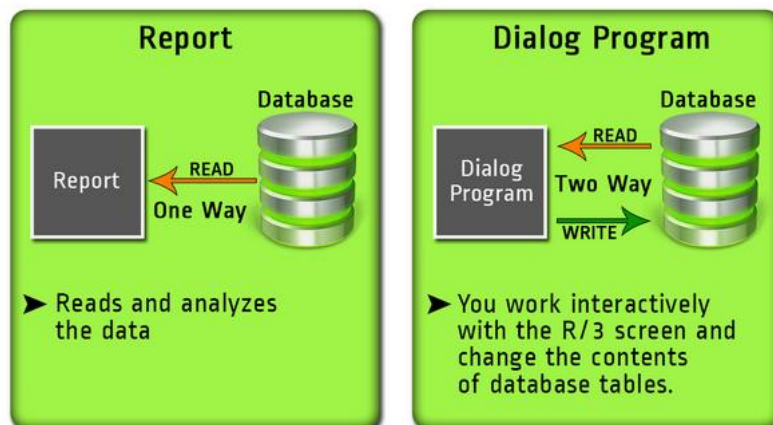
If your ABAP program demands user input, Dialog programming is used.

A user dialog is any form of interaction between the user and the program and could be any of the following

- Entering data
- Choosing a menu item
- Clicking a button
- Clicking or double clicking an entry

Below is the difference of Report and Module pool Programs.

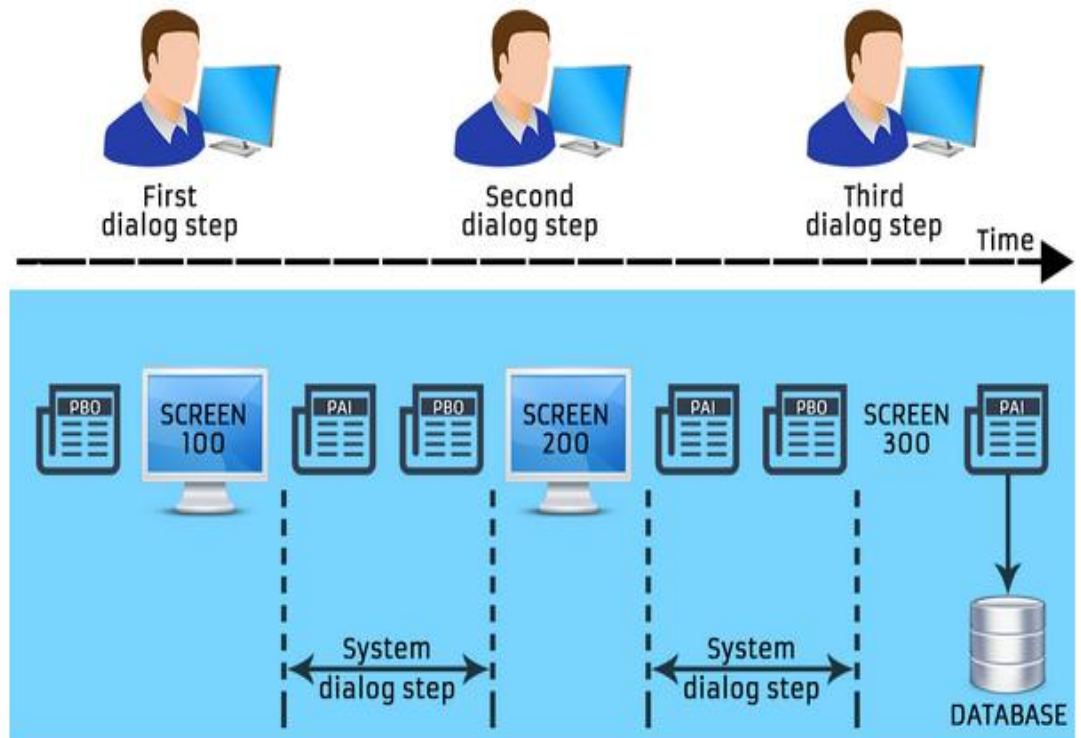
Difference between Report and Dialog Programs



Flow of program in Module Pool:

Dialog programming may have many screen which interact with each other via events. Below is the abstract detail picture of the flow:

Process Flow for a Dialog Program



Components of Module Pool programs:

Transaction code

- The transaction code starts a screen sequence.
- You create transaction codes in the Repository Browser in the ABAP Workbench or using Transaction SE93.
- A transaction code is linked to an ABAP program and an initial screen.
- You can start a screen sequence from any ABAP program using the CALL SCREEN statement.

Screens

- Each dialog in an SAP system is controlled by one or more screens.
- You create screens using the Screen Painter in the ABAP Workbench through transaction SE51
- Each screen belongs to an ABAP program.
- These screens consist of a "screen mask" or "layout" and its flow logic. The screen has a layout that determines the positions of input/output fields and other graphical elements such as checkboxes and radio buttons.
- Flow logic determines the logical processing within screen.

GUI status

- Each screen has a GUI status(es) which are independent components of a program.
- This controls the *menu bars, standard toolbar, application toolbar*, with which the user can choose functions in the application.
- You create them in the ABAP Workbench using the Menu Painter.



ABAP Program

- Each screen and GUI status in the R/3 System belongs to one ABAP program.
- The ABAP program contains the dialog modules that are called by the screen flow logic, and also process the user input from the GUI status.
- ABAP programs that use screens are also known as dialog programs.

Screen Flow Logic with events:

Screen Flow logic is primarily divided into four components.

- Process Before Output (**PBO**) event: which is processed before the screen is displayed
- Process After Input (**PAI**) event: which is processed after a user action on the screen
- Process on help request (**POH**): which is processed when F1 is pressed
- Process on value request (**POV**): which is processed when F4 is pressed

Dynpro

- A screen together with its Flow logic is called a Dynpro ("Dynamic Program" since the screen flow logic influences the program flow)
- Each dynpro controls exactly one step of your Dialog Program.
- The screens belonging to a program are **numbered**. The screen flow sequence can be either linear or cyclic. From within a screen chain, you can even call another screen chain and, after processing it, return to the original chain. You can also override the statically-defined next screen from within the dialog modules of the ABAP program.

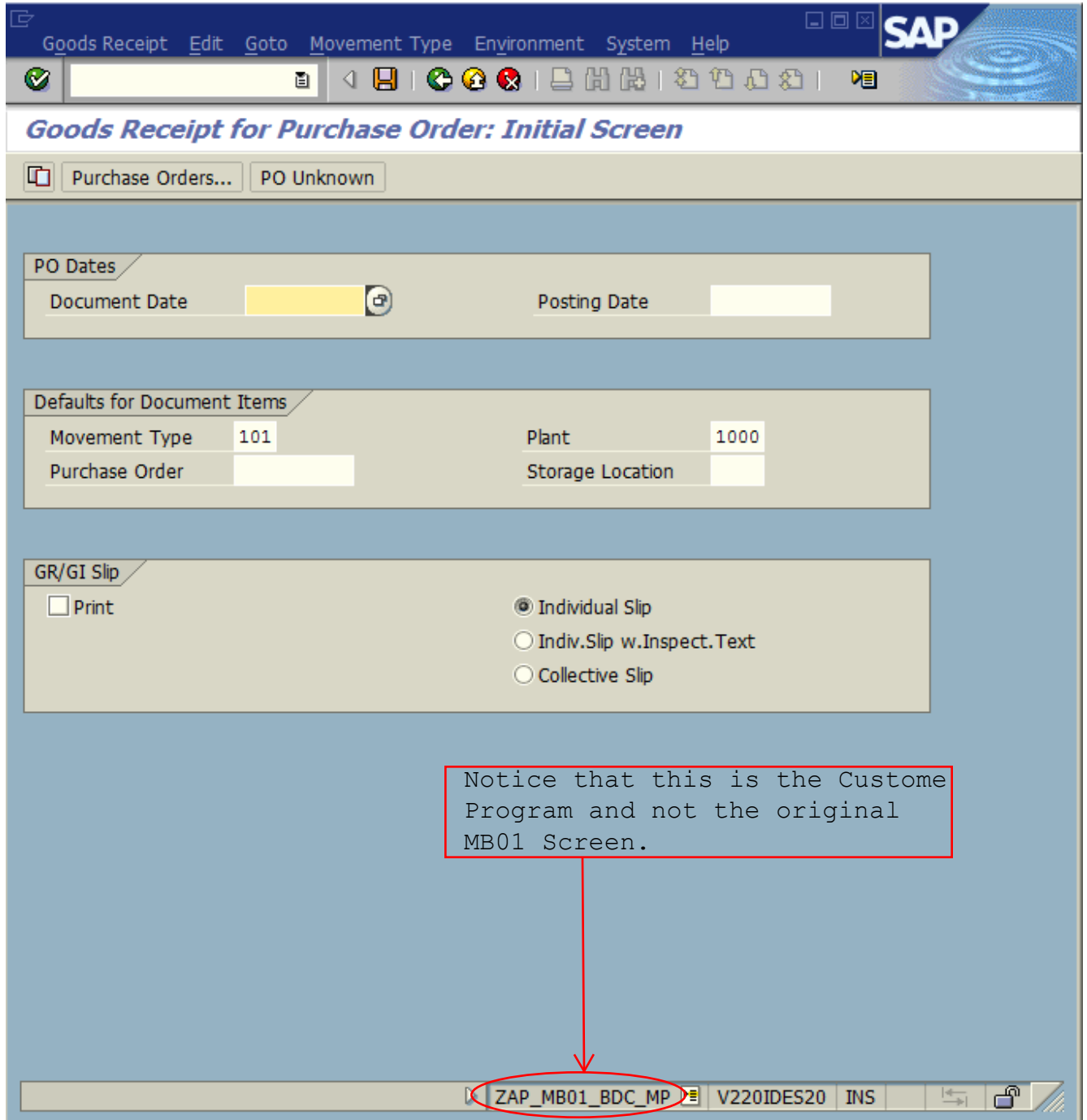


ABAP Module Pool

- On a PBO or PAI event a Dynpro calls an ABAP dialog program. Collection of such programs is called the ABAP module pool.
- For example modules called at the PAI event are used to check the user input and to trigger appropriate dialog steps, such as the update task.
- All dynpros to be called from within **one** transaction refer to a common module pool.



GUI of ZMB01 transaction Module Pool:



Interview questions on Module Pool:

1. What is PBO and PAI events?

PBO- Process Before Output-It determines the flow logic before displaying the screen.

PAI-Process After Input-It determines the flow logic after the display of the screen and after receiving inputs from the User.

2. What is dynpro? What are its components ?

A Dynpro (Dynamic Program) consists of a screen and its flow logic and controls exactly one dialog steps.

The different components of the Dynpro are :

Flow Logic: calls of the ABAP/4 modules for a screen .

Screen layout: Positions of the text, fields, push-buttons and so on for a screen

Screen Attributes: Number of the screen, number of the subsequent screen, and others

Fields attributes: Definition of the attributes of the individual fields on a screen.

3. What is a ABAP/4 module pool?

Each Dynpro refers to exactly one ABAP/4 dialog program. Such a dialog program is also called a module pool, since it consists on interactive modules.



4. What is GUI status? How to create /Edit GUI status?

A GUI status is a subset of the interface elements used for a certain screen. The status comprises those elements that are currently needed by the transaction. The GUI status for a transaction may be composed of the following elements:

- Title bar.
- Menu bar.
- Application tool bar
- Push buttons.

To create and edit GUI status and GUI title, we use the Menu Painter.

5. How does the interaction between the Dynpro and the ABAP/4 Modules takes place?

-A transaction is a collection of screens and ABAP/4 routines, controlled and executed by a Dialog processor. The Dialog processor processes screen after the screen, thereby triggering the appropriate ABAP/4 processing of each screen. For each screen, the system executes the flow logic that contains the corresponding ABAP/4 processing. The controls pass from screen flow logic to ABAP/4 code and back.

6. How does the Dialog handle user requests?

-when an action is performed, the system triggers the PROCESS AFTER INPUT event. The data passed includes field screen data entered by the user and a function code. A function code is a technical name that has been allocated in a screen Painter or Menu Painter to a many entry, a push button, the ENTER key or a function Key of a screen. An internal work field (ok-code) in the PAI module evaluates the function code, and the appropriate action is taken.



7. What is to be defined for a push button fields in the screen attributes?

-A function code has to be defined in the screen attributes for the push buttons in a screen.

8. How are the function code handles in Flow Logic?

– When the User selects a function in a transaction ,the system copies the function code into a specially designated work field called OK_CODE.This field is Global in ABAP/4 Module Pool. The OK_CODE can then be evaluated in the corresponding PAI module. The function code is always passed in Exactly the same way , regardless of Whether it comes from a screen’s pushbutton, a menu option ,function key or other GUI element.

9.What controls the screen flow?

- The SET SCREEN and LEAVE SCREEN statements controls screen flow.



Problem

SCENARIO: Goods Receipt (GR) in bulk are to be manually carried out, as a simple solution we develop a MODULE POOL program with the background processing of BDC – Call transaction method.

We call a user defined transaction code ZMB01 (INSTEAD OF MB01/MIGO) for bulk GRN, Further we push the data to MB01 using Call transaction method of BDC.

We designed 1 screen instead of two screens like of MB01, keeping the process simple we store the input records and later pass them to BDC.



SOLUTION

Proposed a transaction for module pool program as ZMB01 which receives the input from user for the incoming document (like for eg: Purchase Order) and later push it to the Batch Data Conversion (BDC) with help of Call transaction method for posting the incoming documents to system.

Enjoy Coding SAP 😊

Regards,

Pavan Golesar.

SAP ABAP CERTIFIED Consultant.

© abaper.weebly.com

[SAP University Alliance | Pavan Golesar \(sapparamount@gmail.com\)](mailto:sapparamount@gmail.com)

