BDC – Batch Data Communication.

Feeding data into the SAP system but huge volumes of data. SAP ABAP programs and some standard functionalities make this happen.

It is used in case of migration projects or when a manual system or an old legacy system moves to SAP R/3 architecture.

2 scenarios can take place:

1. Transfer of data happens once and for all
2. Transfer of data happens regularly

Programmer can run the transaction (used to transfer the data) at once or schedule it to run at a later time.

2 way Data Transfer:

1. Inbound (data comes from external system or another SAP system into SAP system )
2. Outbound( data goes out to external system or another SAP system)

3 Methods to do the data transfer: (1 and 2 are known as Batch Input Methods)

1. Session Method
2. Call Transaction Method
3. Direct Input (standard programs)

**Enjoy sap transactions** are the transaction where u can find all the screens like header data and the item data all in one page only i.e all in one screen itself.  
  
Enjoy transaction code generally ends with N  
  
Ex..  
  
ME21N  
  
VL01N  
  
VL31N

SAP Enjoy transactions ( They generally end with N) have random screens and pop-ups and the flow of screens could be different  
  
BDC cant be used. BAPI's should be used for any processing required programmatically.  
  
Enjoy transactions uses OOPS ABAP code using classes and Methods.  
  
Oridnary Transactions have predefined screens and the flow/sequence of screens executed is defined.  
  
BDC can be used and the flow of screens is predefined and always the same.

Difference between Call Transaction and Session Method

|  |  |
| --- | --- |
| **Call Transaction** | **Session Method** |
| 1 Transaction at a time | Several transactions can be processed together. Hence the preferred method.( obviously depending on the method) |
| Data is processed immediately. Hence the faster process | Not processed immediately. They are placed in batches for execution. |
|  | SM35 Tcode to see the results of batch job |
| No sessions are created | Sessions are created |
| Error handling has to be explicitly handled | Implicitly handled |
| For lesser records like (few hundreds of data) | For huge volume of data - thousands |
| Asynchronously | Synchronously – it waits for |

Steps of session method:

|  |
| --- |
| 1. bdc\_open\_group |
| 1. loop |
| 1. bdc\_insert |
| 1. endloop |
| 1. bdc\_close\_group |

1 ) BDC\_OPEN\_GROUP ; Open batch input session for adding transactions

Function Module BDC\_OPEN\_GROUP

Allows you to open a batch input session when it is created.

**Parameters:**

* Required entries:
  + CLIENT: Client
  + GROUP: Folder name ( session name u want to create in SM35)
  + USER: Batch input authorization user
* Optional:
  + KEEP: Flag for retaining processed folders/ successful logs of the session can be seen
  + HOLDDATE: Lock folder for processing until specified date
  + RECORD: Flag for BI recording 'X' oder ' '
  + DEST: (not applicable)
  + PROG: Creating program
  + DCPFM: Decimal character used in the folder data
  + DATFM: Date format used for the data display
* Return:
  + QID: Unique database key

Close the batch input session with function module BDC\_CLOSE\_GROUP

BDC\_CLOSE\_GROUP needs no parameters.

It automatically closes the session that is currently open in the program.

A session must be closed before another session is open from the same program.

A session cannot re-opened once it has been closed.

**A new call to BDC\_OPEN\_GROUP with the same session name creates a new session with the same name**.

If this group is not used then

CALL TRANSACTION 'MM01' USING it\_bdcdata MODE 'A' UPDATE 'S'.

Mode A means All screen  
Mode E means only Error screens  
Mode N means No screen

And Updation is Synchronously.

 it\_bdcdata-dynbegin = 'X'."continue button ?? --- in call transaction prog

notepad has to be tab separated else it will override the data in the output.