Capgemini

ABAP Part III

Lesson 01: Batch Data Communication

Lesson Objectives

After completing this lesson, participants will be able to -

- Know the different Data Transfer Methods
- Use BDC Data Transfer
  - Session Method
  - Transaction Method

Introduction

Data is transferred from an external system into the SAP R/3 System

Data Transfer is used when,

- Transfer data from an external system into an R/3 System as it is installed.
- Transfer data regularly from an external system into an R/3 System.

SAP applications support the data transfer of numerous SAP business objects

The data transfer program specifies the data format definition that is necessary to import the data into the R/3 System.

Once the data has been exported it can be imported into the system using a generated data transfer program.

BDC

Programming technique for loading data into SAP

BDC is a combination of ABAP/4 Programming and built in SAP functionality

Simulates the act of a user entering data into an SAP transaction.

The system picks the data from Database using ABAP/4 program and feeds it to an SAP system using the corresponding transaction screen by screen

The programmer can choose whether the transactions should run immediately or at a later time

Why BDC?

When a company decides to implement the SAP R/3 to manage business-critical data, it usually does not start from a no-data situation

A SAP R/3 project comes in to replace or complement to an existing application.

In the process of replacing current application and transferring application data, two situations might occur:

- The first is when application data to be replaced is transferred at once and only once.
- The second situation is to transfer data periodically from external system to SAP and Vice versa.

Why BDC?

It is very difficult to transfer large amount of data manually from an external system into the R/3 System.

A data transfer method is required that transfers the data automatically in the background without user intervention.

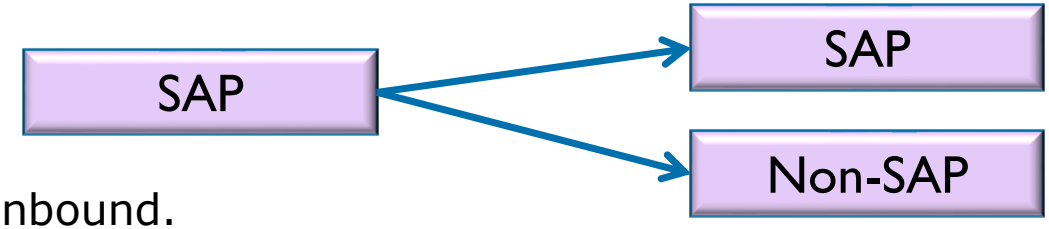
BDC is a very good answer to this problem.

Using BDC the data can be transferred both at once and periodically depending on the users' requirement.

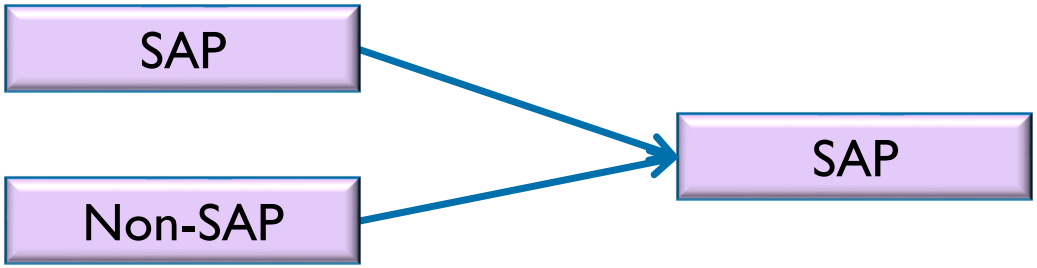
BDC

The data transfer can be two- way

- Outbound
  - Outbound is a data transfer from SAP to SAP / Non SAP.



- Inbound.
  - Data transfer from External System into SAP



Methods for transferring Data

There are three ways to transfer the data into SAP from any external system.

- SESSION METHOD
- CALL TRANSACTION
- DIRECT INPUT.

SESSION Method and CALL TRANSACTION are called as BATCH INPUT Methods

Direct Inputs are standard programs to post the data into SAP.

Advantages of Batch Input

Processes large data volumes in batch

Can be planned and submitted in the background

No manual interaction is required when data is transferred.

Data integrity is maintained as whatever data is transferred to the table is through transaction .Hence batch input data is submitted to all the checks and validations.

Steps involved in writing a Data Transfer Program

Analyzing Transaction

Declaring Internal Tables

Transferring Data From Flat Files into internal Tables

Population of BDCDATA into Internal Table

Looping internal table to any of the data transfer methods

Analyzing Transaction

The following steps are involved in Analyzing transaction

- The name, Type and length of the field
- Screen number and name of the module pool program
- Determine the mandatory fields
- Determine the fields which accepts standard values
- Determine if data conversion is required when converting to SAP format

Declaring Internal Tables

Three internal tables to be declared

- Internal table with structure similar to
  - The flat file
  - DDIC structure BDCDATA
  - DDIC Structure BDCMSGCOLL

Transferring Data from flat file to Internal Table

Depends on the source of the file

The flat file can be in

- Presentation Server
- Application Server

Depending on the server , the programming is done in ABAP to upload data from flat file to internal table.

Transferring Data from flat file to Internal Table

If the flat file resides on the presentation server the function module GUI_UPLOAD is used.

The important parameters of GUI_UPLOAD are

- Filename
- FileType
- Has_field_seperator
- Data_tab

File residing on the Application Server

If the file is residing on the application server, the following statements are used :

- OPEN DATASET for opening files
  - OPEN DATASET <dsn> [Additions].
  - OPEN DATASET <dsn> FOR INPUT.
- CLOSE DATASET for closing files
  - CLOSE DATASET <dsn>.
- DELETE DATASET for deleting files
  - DELETE DATASET <dsn>.
- READ DATASET <dsn> INTO <f> [LENGTH <len>] for Reading Data from Files
- TRANSFER <f> to <dsn> [LENGTH <len>] for writing Data to Files

Population of BDCDATA into the Internal Table

After uploading data to internal table, the BDCDATA internal table is filled with values required to process single record .

Loop

Loop this internal table and employ any of the below mentioned methods of data transfer

- CALL TRANSACTION
  - CALL TRANSACTION <tcode> USING IT_BDCDATA MESSAGES INTO IT_BDCMSGCOLL
- SESSION METHOD
  - Insert the IT_BDCDATA into a session by calling the Function Module - BDC_INSERT and is processed
- DIRECT INPUT

CALL TRANSACTION METHOD

The Conversion program uses the ABAP statement CALL TRANSACTION USING to run an SAP transaction

External data does not have to be deposited in a session for later processing.

The entire batch input process takes place inline in the program.

Processing batch input data with CALL TRANSACTION USING is the faster of the two recommended data transfer methods.

Legacy data is processed inline in the data transfer program.

Call Transaction Method (Contd.)..

CALL TRANSACTION <tcode>
USING <bdc_tab>
MODE <mode>
UPDATE <update>
MESSAGES INTO <BDCMSGCOLL_TAB>.

- Where

<tcode> : Transaction code.
<bdc_tab> : Internal table of structure BDCDATA.
<mode> : Display mode.
<update>: Update mode.
<BDCMSGCOLL_TAB> : Internal table of structure BDCMSGCOLL.

BDCMSGCOLL

BDCMSGCOLL has the following structure:

Field Name	Description
TCODE	BDC Transaction Code
DYNAME	Batch input module name
DYNUMB	Batch input screen number
MSGTYP	Batch input message type
MSGSPRA	Language ID of a message
MSGID	Batch input message ID
MSGNR	Batch input message number
MSGV1	Variable part of a message
MSGV2	Variable part of a message
MSGV3	Variable part of a message
MSGV4	Variable part of a message
ENV	Batch input monitoring activity
FLDNAME	Field name

Structure of the BDC Table

Field	Type	Length	Description
Program	CHAR	40	Program name of the transaction
DynPro	NUMC	4	Screen number of the transaction
DynBegin	CHAR	1	Indicator for new screen
Fnam	CHAR	132	Name of the database field from screen
Fval	CHAR	132	Value to submit to field

BDC

After the BDC table has been built, it has to be submitted for SAP Processing

BDC - Display Mode

Specifies whether the whether data transfer processing should be displayed as it happens.

There are 3 display modes to be chosen from:

- 'A' - stands for 'Display all'. All screens and the data that goes in them appear when we run the program.
- 'N' - stands for 'No display'. All screens are processed invisibly, regardless of whether there are errors or not. Control returns to the program as soon as transaction processing is finished.
- 'E' - stands for 'Display errors only'. The transaction goes into display mode as soon as an error in one of the screens is detected. The errors can then be corrected .

BDC - Update Mode

Specifies how updates produced by a transaction should be processed

3 update modes are available.

- A’ - ‘Asynchronous updating’.
  - The called transaction does not wait for any updates it produces, to be completed.
  - Results in faster execution of the data transfer program.
  - ’Asynchronous processing’ is NOT recommended for processing any larger amount of data because the called transaction receives no completion message from the update module in asynchronous updating.
  - The calling data transfer program, in turn, cannot determine whether a called transaction ended with a successful update of the database or not.

BDC - Update Mode

'S' - 'Synchronous updating'.

- The called transaction waits for any updates that it produces to be completed.
- Execution is slower than with asynchronous updating because called transactions wait for updating to be completed.
- The called transaction will return any update error message that occurs to the program.
- It is much easier to analyze and recover from errors.

- 'L' - 'Local updating'.

- If the data is updated locally, the update of the database will not be processed in a separate process, but in the process of the calling program.

The Messages Parameter

When the records are uploaded in database table by Session Method error record is stored in the log file.

In Call transaction there is no such log file available and error record is lost unless handled.

The MESSAGES specification indicates that all system messages issued during a CALL TRANSACTION USING are written into the internal table < BDCMSGCOLL_TAB >

Return Codes:

Values	Explanation
0	Successful
<=1000	Error in Dialog Program
> 1000	Batch Input Error

Session Method

The second way of data transfer is by submitting the BDC session to the system for batch processing.

Several transactions can be processed together

Unlike Call Transaction data is not processed immediately

It’s placed into the SAP batch queue for later processing.

There is a transaction as SM35 which allows the user to view the results of a batch job that has been processed by the system.

Session Method (Contd.).

A session records transactions and data in a special format that can be interpreted by the R/3 System.

The data that a session enters into transaction screens is subject to the same consistency checking as in normal interactive operation.

Batch input sessions are subject to the user-based authorization checking that is performed by the system.

BDC Function Modules

There are three functional modules to be called from BDC Program for submitting the transactions for processing

- BDC_OPEN_GROUP: for creating session

- BDC_INSERT: Transferring data from internal table(BDCDATA) to session.

- BDC_CLOSE_GROUP: Closing session

Important Aspects of Session Interface

Asynchronous processing

Transfers data for multiple transactions

Synchronous database update

A batch input processing log is generated for each session

Sessions cannot be generated in parallel.

The batch input program must not open a session until it has closed the preceding session.

Steps to Work With Session Method

- Generate the batch input session using function module BDC_OPEN _GROUP.

- BDC_OPEN_GROUP has the following Export Parameters:
  - CLIENT
    - Client in which the session is to be processed
  - GROUP
    - Name of the session to be created
  - HOLDDATE
    - The session is locked and may not be processed until the date specified
  - KEEP
    - Retains session after successful processing
  - USER
    - The user name that is used for checking authorizations if a session is started in background processing

Transaction of the Session

For each transaction of the session

- Enter the value for all screens and fields in the BDC data structure, that must be processed in the transaction (i.e. fill the internal table IT_BDCDATA).
- Use function module BDC_INSERT to transfer the transaction and the IT_BDCDATA to the session.
- BDC_INSERT has the following Parameters
  - TCODE
    - The transaction code to be run
  - POST_LOCAL
    - Parameter to update data locally.
  - DYNPROTAB
    - The BDCDATA structure that contains the data that is to be processed by the transaction.

Close the batch input session

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.

Processing the Session

Process the Session Online through SM35 in Background through program RSBDCSUB.

- Repeat from the Step 2, for each transaction, when multiple transactions has to be processed through the same Session.

Processing Session

A batch input session is a set of one or more calls to transactions along with the data to be processed by the transactions.

The system normally executes the transactions in a session non-interactively, allowing rapid entry of bulk data into an R/3 System.

A Session can be processed in two ways:

- Automatically
- Explicitly

Processing Sessions Automatically

In most cases, batch input sessions can be processed automatically.

It is not necessary for a session to wait until a system administrator explicitly starts the processing of the session.

The ABAP program RSBDCSUB must be scheduled as a periodic job in the R/3 background processing system.

RSBDCSUB checks for and starts any batch input sessions that has not yet been run.

It schedules such sessions for immediate execution in the background processing system.

Procedure

Schedule ‘RSBDCSUB’ to run periodically in one or more background jobs.

If there are batch input scheduled to run regularly, separate jobs can be scheduled for each of the scheduled data transfers.

The start time for the RSBDCSUB job can be set according to the batch input schedule.

Variants can be used to restrict RSBDCSUB only to the batch input sessions that is expected.

Input For the Program RSBDCSUB:

Session name

Date and time of generation

Status : ready to run or held in the queue because of errors

Starting Sessions Explicitly: Run Modes

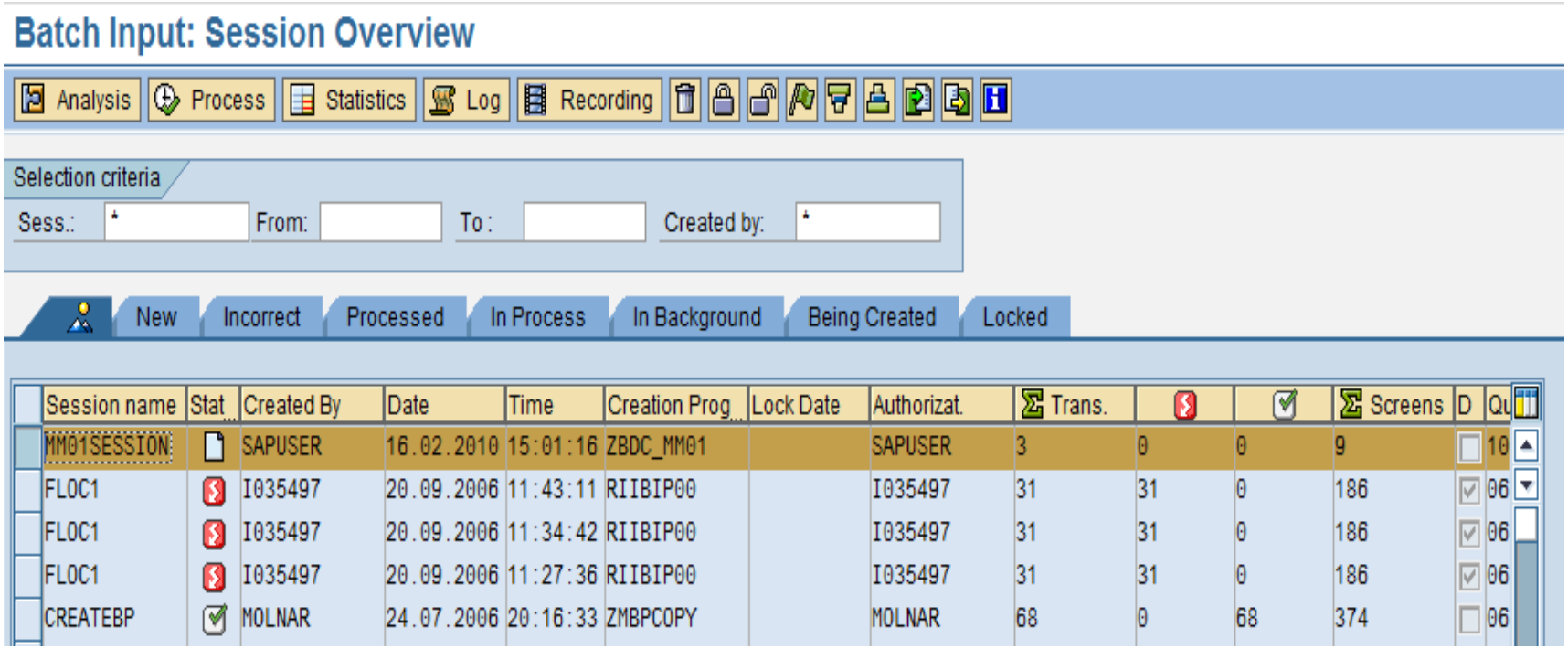
Running a batch input session executes the transactions in the session and enters data into an R/3 System.

Usually, the system will run batch input sessions automatically. However, it can also be started manually for the following reasons:

- To correct transactions that had errors
- To check that the transactions in a session have been generated correctly by running the first several transactions
- To start a session on special request (the session would not be started automatically or must be started right away).

Prerequisites

- Start the batch input management tool (SM35):



Procedure

To start the session, mark the session and choose PROCESS from the toolbar

Choose how to run a session and with what logging and display options

Run Modes

- There are 3 ways to run a session
  - Process/Foreground
  - Display Errors only
  - Background

BDC

- Process/foreground:
  - Transactions that contain errors can be corrected and step through transactions that have not yet been executed.

- Display errors only
  - This mode is like Process/foreground except that transactions that have not yet been run and which do not contain errors are run non-interactively.
  - If an error occurs, processing stops and the screen upon which the error occurred is displayed.
  - With Process foreground or Display errors only mode, transactions that have the status Incorrect can be restarted

- Background:
  - This mode is used to schedule a session for immediate processing in the background processing facility.

Displaying Session Logs

- The batch input system keeps a detailed log of each session that is processed.
- The log contains not only progress messages from the batch input system itself, but also error messages from the transactions that are processed.
- To analyze an error in a session, start checking the session log for relevant messages.
- A session log is kept only if the session was generated with the KEEP option or if the session is aborted or contains an error.

Session Status

Sessions in the session queue are sorted by date and time of generation and are grouped in different lists according to their status.

Possible statuses are as follows:

- New:
  - Session was created and recorded but not yet processed.
- Incorrect :
  - Held in the session queue because of errors in transactions (Errors in sessions)
  - Transactions that contain errors are aborted
  - All correct transactions are processed.
  - A session can be restarted to correct the erroneous transactions with one of the interactive execution modes offered by the batch input system.

Status

- Processed:
  - It refers to all those sessions that have been successfully run.
  - The log generated can be displayed by the session.
  - All completed sessions generate a log.
  - A completed session cannot be run a second time.
  - Note:
    - Only sessions that were generated with the KEEP option are held in the queue after processing.
    - Other sessions are deleted after they are successfully completed.
- In processing:
  - This status is seen only if the queue is displayed while a session is being run.

Status (Contd.).

- In Background:
  - This status appears only if the Session is processed in the Background Mode
- Being Created:
  - This status is seen only if the queue is displayed while a session is being generated
- Locked:
  - Status when the session is locked

Direct Input Method

Direct input is a method for automatically transferring large quantities of data to the SAP R/3 System without the need for online processing.

Data can be transferred at the time of installation and also at a later point of time

Direct input can be used at different times for different purposes:

- To transfer existing data when installing the SAP R/3 System
- To change current data when the SAP R/3 System is in operation

Direct Input Method

To ensure data consistency, the data transferred automatically must undergo the same checks as the data entered manually.

- The same checks are made.
- The same error messages and warnings are issued.
- The data is posted and updated in the database in the same way.

The direct input program used for material master records is RMDATIND.

- It performs the following functions:
  - Creating material master records MM01
  - Changing material master records MM02

Stock data from Inventory Management is transferred to material master records using batch input program RM07MMBL.

Direct Input Method

To enhance the batch input procedure, the system offers the direct input technique, especially for transferring large amounts of data.

In contrast to batch input, this technique does not create sessions, but stores the data directly.

It does not process screens.

To enter the data into the corresponding database tables directly, the system calls a number of function modules that execute any necessary checks.

In case of errors, the direct input technique provides a restart mechanism.

However, to be able to activate the restart mechanism, direct input programs must be executed in the background only.

To maintain and start these programs, use program RBMVSHOW or Transaction BMV0.

Direct Input Method - Example

Program	Application
RFBIBL00	FI
RMDATIND	MM
RVAFSS00	SD
RAALTD11	AM
RKEVEXT0	CO-PA

Summary

In this lesson, you have learnt:

- The different Data Transfer Methods
- How to use BDC Data Transfer
  - Session Method
  - Transaction Method

Review Question

Question 1: The ABAP program _____ can be scheduled as a periodic job in the R/3 background processing system.

Question 2: CALL TRANSACTION USING is the faster than Call session and Direct Input method of data Transfer.