IMS DB ONLINE TRAINING – DAY 9

(c) Ez-Learn Global Pvt Ltd, India

IMS LOGGING FACILITIES

When IMS is executing via online or batch, it provides a number of features that ensure data integrity or aid in the restart ability, or recoverability, of IMS data bases and application programs. These features include the following:

- Commit or synchronization points
- Logging
- Back out
- Restore/recovery
- Generalized Sequential Access Method
- Checkpoints
- Extended restart

Each of these features contributes to the integrity of IMS as an excellent data base management system (DBMS).

(c) Ez-Learn Global Pvt Ltd, India

IMS LOGGING FACILITIES

IMS performs two functions to provide data base integrity:

- It allows only one program to process a given segment and its dependent segments in a data base record at a time. *This is called program isolation.*

- It prevents your program from accessing segments that have been deleted, replaced, or inserted by other programs until a commit point has been reached.

Commit or Synchronization Points:

A commit point, also called a synchronization point, is a point at which a unit of work is complete, and IMS makes all data base changes permanent. The data is then made available to other programs for processing.

Commit points can occur at the following times:

- If the program terminates normally.

- If the program issues a checkpoint call (this will be discussed later).

- If the program completes a unit of work, such as the retrieval of a message from the message queue. This is performed automatically in most online programs (MPP) and transaction-oriented BMPs.

(c) Ez-Learn Global Pvt Ltd, India

Usage of GSAM

GSAM databases are sequentially organized databases designed to be compatible with MVS data sets either PS or ESDS

GSAM databases can be on a data set previously created or one later accessed by the MVS access methods VSAM or QSAM

GSAM is useful if you need an application program that accesses MVS data sets to use the IMS symbolic checkpoint call

GSAM databases are loaded in the order in which you present records to the load program. You cannot issue DLET and REPL calls against GSAM databases; however, you can issue ISRT calls after the database is loaded but only to add records to the end of the data set. Records cannot be randomly added to a GSAM data set.

(c) Ez-Learn Global Pvt Ltd, India

Usage of GSAM

GSAM data bases have the following restrictions when used in IMS programs:

- They cannot be used in MPP (online) programs.
- They cannot have a hierarchy. They cannot use keys.
- They cannot use the DLET and REPL calls.
- Records can only be added at the end of the file.

The following are advantages of GSAM data bases:

- They can be created and accessed as OS/VS sequential files in non- DL/I programs.
- They can be checkpointed.
- They can be restarted using the XRST call.

Because GSAM file participate in the checkpoint and restart facilities, sequential files are converted to GSAM files for BMP processing.

(c) Ez-Learn Global Pvt Ltd, India

Usage of GSAM

The steps required for converting a sequential data set into a GSAM data base are outlined in the following:

1. Create a DBD for the data base, as shown below:-

```
DBD    NAME=dbdname,ACCESS =(GSAM, BSAM or VSAM)
DATASET        DD1=ddname,RECFM =  ,RECORD=(max, min)
DBDGEN
FINISH
END
```

2. Include the GSAM PCB in the program's PSB as shown below:-

```
PCB    TYPE =DB.,……..
PCB    TYPE = GSAM,NAME = dbdname,PROCOPT=G(S) or L (S)
 PSBGEN       PSBNAME =
END
```

Note : GSAM PCBs must appear last in the PSB.

www.ezlearnglobal
(c) Ez-Learn Global Pvt Ltd, India

Usage of GSAM

After converting sequential files into GSAM data bases, you must change the application program to replace all OS/VS reads and writes to GSAM DL/I calls. You can use one of five basic calls against a GSAM data base:

OPEN – OPEN permits the explicit opening of the data base. It is not required because IMS automatically opens the data base the first time a get or insert call is issued.

CLSE – CLSE allows the explicit closing of the data base. It is not required because IMS automatically closes the data base upon termination of the program.

GU – GU permits the retrieval of a specific record in the database. Because GSAM does not allow segments or keys, you retrieve a record by using record search argument (RSA), which will be discussed later.

GN – GN allows the retrieval of the next sequential record in the data base.

ISRT –ISRT permits the insertion of a new record in the data base.

However, you can only add to the end of the data base.

(c) Ez-Learn Global Pvt Ltd, India

Checkpoint-Restart Logic

In addition to the log file, back up and recover utilities, another important IMS tool providing restart ability to IMS programs is the check pointing facility.

The IMS check pointing facilities are available to all programs whose PSB specifies **CMPAT = YES**. This is because any IMS program issuing checkpoint calls must do so through the input/output program communication block (I/O PCB).

ENTRY 'DLITCBL' USING **IO – PCB**, DB1 –PCB,...DBN – PCB.

I/O PCB pointer specification in program code

Additionally, you must code an I/O area used for the I/O PCB.

(c) Ez-Learn Global Pvt Ltd, India

Checkpoint-Restart Logic

There are two types of Checkpoint logic methods as follows:-

a) Basic Checkpoint Method

b) Symbolic Checkpoint Method

a) Basic Checkpoint Method:

The Basic method of check pointing invokes IMS'S standard checkpoint functions. While all programs can use the Basic method, it is the only checkpoint method available to MPP'S (Online programs). The application program must provide its own method of restarting, because the extended restart call is not supported. Neither OS/VS nor GSAM files are supported with the Basic checkpoint method.

(c) Ez-Learn Global Pvt Ltd, India

Checkpoint-Restart Logic (Basic)

```
01 CHKP-ID.
   05   FILLER          PIC     X(4)    VALUE   'CT7P'.
   05   CHKP-NUM        PIC     9(4)    VALUE   0.

ADD 1 TO CHKP-NUM.

CALL    'CBLTDLI'       USING   C-CHKP-FUNC,
                                IO-PCB,
                                CHKP-ID.
```

It is our responsibility to provide an 8-character checkpoint ID and to increment it as processing dictates. Once we issue the checkpoint call, the only valid status code is blank. A message is sent to the master console, as well as to the JES log, with the checkpoint ID and a date and time stamp. You will need this information, if a checkpoint restart becomes necessary.

(c) Ez-Learn Global Pvt Ltd, India

Checkpoint-Restart Logic (Basic)

Basic Checkpoint Summary

The following is true when using the Basic checkpoint method:-

- Used in DL/I batch, BMP, and MPP
- Releases all held IMS resources
- Makes all data base changes (ISRT'S, REPL'S, and DLET'S) permanent
- Destroys position in all non-GSAM data bases
- Programmer must provide restart logic to restore variables and data base position

(c) Ez-Learn Global Pvt Ltd, India

Checkpoint-Restart Logic (Symbolic)

b) Symbolic Checkpoint Method

Only BMP and batch programs can use this method. Up to seven I/O areas can be saved with Symbolic checkpointing. Symbolic method is preferable to the Basic checkpoint method. Symbolic checkpointing uses the extended restart call to restart an abended job.

```
01 IO_AREA_LTH          PIC S9(9) COMP VALUE +nnn.      Longest segment  or Segment path
01 CHKP-ID.
     05 FILLER          PIC  X(06) VALUE `CT7???'.
     05 CHKP-NUM        PIC  9(02) VALUE 0.

 01 CHKP-LTH1           PIC S9 (09) COMP VALUE +nnn.
 01 CHKP-AREA1.

    05 FIELDI....
```

Up to 7 areas

(c) Ez-Learn Global Pvt Ltd, India

Checkpoint-Restart Logic (Symbolic)

```
ADD 1 TO CHKP-NUM

CALL    'CBLTDLI'       USING   CHKP-FUNC,
                                IO-PCB,
                                IO-AREA-LTH,
                                CHKP-ID,
                                CHKP-LTH1,
                                CHKP-AREA1,
                                CHKP-LTH7,
                                CHKP-AREA7.
```


(c) Ez-Learn Global Pvt Ltd, India

Checkpoint-Restart Logic (Symbolic)

IMS Extended Restart

Let us see how to restart and abended job that has Symbolic checkpoint method.

If a batch program abends *and dynamic back out was not used*, you must run the back out utility. This restores your IMS data bases to the last checkpoint, or to a checkpoint specified in the back out utility. Next, you must create restart JCL that specifies the checkpoint from which to restart. This JCL (shown in the following) is basically the same JCL used for batch IMS execution. The differences are the addition of the checkpoint ID in the parm, and the additional DD IMSLOGR.

```
//USER011      JOB              . . .
//STEPnnn      EXEC    PGM=DFSRRC00,
//              PARM='BMP,pgmname,psbname,.....chkptid'
                     or
//              PARM='DLI,pgmname,psbname,......chkptid'
//STEPLIB      DD   DSN=authoriz.ims.library,DISP=SHR
//             DD   DSN=system.loadlib,DISP=SHR
//IMSLOGR      DD   DSN=log.file(0),DISP=OLD
//IEFRDER      DD   DSN=log.file(+1),DISP=(NEW,KEEP),
//     UNIT=dasd,DCB=(RECFM=VB,LRECL=4148,BLKSIZE=4152)
```

www.ez-learn.global
(c) EZ-Learn Global Pvt Ltd, India

Checkpoint-Restart Logic (Symbolic)

```
01 IO-AREA-LTH          PIC S9(9) COMP          VALUE +nnn.
01 XRST-ID              PIC X(12).
01 CHKP-LTH1            PIC S9(9) COMP VALUE +nnn.
01 CHKP-AREA1.
   05 FIELD1.....

MOVE SPACES    TO       XRST-ID.
CALL 'CBLTDLI'          USING   XRST-FUNC,
                                L-IO-PCB,
                                IO-AREA-LTH,
                                CHKP-AREA1,
                                •
                                •
                                CHKP-LTH7,
                                CHKP-AREA7.
```


(c) Ez-Learn Global Pvt Ltd, India

Checkpoint-Restart Logic (Symbolic)

```
IF XRST-ID = SPACES
   PERFORM NORMAL-START-ROUTINE
ELSE
   PERFORM RESTART-ROUTINE.
```

You should have only one XRST call in your program, and it should be the first call. If a restart was requested, the XRST call automatically restores all of the variables you declared in the checkpoint area. It also restores the position in all data bases and GSAM files at the current position when that checkpoint was taken. (IMS does this by issuing internal GU calls).

Symbolic checkpointing is the only method that provides restartability with the restoration of variables and repositioning in all data bases.

(c) Ez-Learn Global Pvt Ltd, India

Checkpoint-Restart Logic (Symbolic)

Symbolic Checkpoint Summary

The following is true of the Symbolic checkpoint method:

- Used in DL/I batch or BMP-cannot be used with MPP.
- Releases all held IMS resources.
- Makes all data base changes (ISRT'S, REPL'S, and DLET'S) permanent.
- Destroys position in all non-GSAM data bases.
- Records programmer specified save areas (up to seven) and data base position in log file.
- Can use the XRST call to restart program.
- Does not checkpoint OS/VS files-must convert them to GSAM.

(c) Ez-Learn Global Pvt Ltd, India

Checkpoint-Restart Logic (Symbolic)

Extended Restart Summary

The following explains the use of the extended restart call:

- If used, it must be the first DL/I call issued by the program, and can only be issued once (regardless of whether the program is being restarted).
- It can only be used with the symbolic checkpoint method. It causes subsequent checkpoint calls to be recognized as symbolic.
- Restores save area variables and data base position (including GSAM) from checkpoint call.

Program can be restarted from:

1. A specific checkpoint ID
2. A date/time stamp
3. The last checkpoint taken (if the program is BMP and the log file is on DASD JCL only)

(c) Ez-Learn Global Pvt Ltd, India

Exercise 9.1 (Test your knowledge)

- A Commit Point is a point at which a unit of work is complete - *(True/False)*
- Database changes are not made permanent when the job abends - *(True/False)*
- A PS or a ESDS file can be used as a GSAM database - *(True/False)*
- We can do checkpoint restart with GSAM database - *(True/False)*
- CMPAT=Yes is mandatory for BMP processing as well as for using Symbolic checkpoint - *(True/False)*
- In Basic checkpointing method, restart procedure should be handled by the programmer - *(True/False)*
- In Symbolic checkpointing method, restart procedure are handled by XRST call - *(True/False)*

(c) Ez-Learn Global Pvt Ltd, India

End of Day 9

(c) Ez-Learn Global Pvt Ltd, India