Session - 5

Queues

Transient Data Control

- Provides application programmer with a queuing facility
- Data can be stored/queued for subsequent internal or external processing
- Stored data can be routed to symbolic destinations
- TDQs require a DCT entry
- Identified by Destination id - 1 to 4 bytes

TDQs

- **Intra-partitioned** - association within the same CICS subsystem

- **Extra-partitioned** - association external to the CICS subsystem

TDQs

- Operations

  Write data to a transient data queue (WRITEQ TD)

  Read data from a transient data queue (READQ TD)

  Delete an intrapartition transient data queue (DELETEQ TD).

WRITEQ TD

- Syntax :

```
EXEC CICS WRITEQ TD
                    QUEUE(name)
                    FROM(data-area)
                    [LENGTH(data-value)]
                    [SYSID(systemname)]
END-EXEC.
```

Conditions: DISABLED, INVREQ, IOERR, ISCINVREQ, LENGERR, NOSPACE, NOTAUTH, NOTOPEN, QIDERR, SYSIDERR

READQ TD

- Syntax :

```
EXEC CICS   READQ TD
                    QUEUE(name)
                    {INTO(data-area) | SET(ptr-ref) }
                    [LENGTH(data-value)]
                    [NOSUSPEND]
END-EXEC.
```

Conditions : DISABLED, IOERR, INVREQ, ISCINVREQ, LENGERR, NOTAUTH, NOTOPEN, QBUSY, QIDERR, QZERO, SYSIDERR

DELETEQ TD

- Deletes all entries in the queue

- Syntax :

```
EXEC CICS DELETEQ TD
                  QUEUE(name)
END-EXEC.
```

Conditions: INVREQ, ISCINVREQ, NOTAUTH,
QIDERR, SYSIDERR

Destination Control Table

- DCT is to register the information of all TDQs

- Destination Control Program (DCP) uses DCT to identify all TDQs and perform all I/O operations.

- DFHDCT is a macro to define intra & extra partition TDQs TYPE=INTRA/EXTRA

- REUSE option specified along with intra partition TDQ tells whether the space used by TDQ record will be removed & reused after it has been read.

Automatic Task Initiation

- Facility through which a CICS transaction can be initiated automatically

```
DFHDCT    TYPE=INTRA
               DESTID=MSGS
               TRANSID=MSW1
               TRIGLEV=500
```

When the number of TDQ records reaches 500, the transaction MSW1 will be initiated automatically

- Applications

  Message switching & Report printing

Temporary Storage Control

- Provides application programmer the ability to store and retrieve data in a TSQ

- Application can use the TSQ like a scratch pad

- TSQs are

  - Created and deleted dynamically
  - No CICS table entry required if recovery not required
  - Identified by Queue id - 1 to 8 bytes
  - Typically a combination of termid/tranid/operid

- Each record in TSQ identified by relative position, called the item number

TSQs

- Operations
  - Write and Update data
  - Read data - Sequential and random
  - Delete the queue

- Access
  - Across transactions
  - Across terminals

- Storage
  - Main - Non-recoverable
  - Auxiliary - Recoverable
  - TST entry required, VSAM file DFHTEMP

TSQs – Typical Uses

- Data passing among transactions
- Terminal Paging
- Report printing

WRITEQ TS

- Syntax :

EXEC CICS **WRITEQ TS**

QUEUE(name)

FROM(data-area)

[LENGTH(data-value)]

[NUMITEMS(data-area) |

ITEM(data-area) [REWRITE]]

[MAIN|AUXILIARY]

[NOSUSPEND]

END-EXEC.

Conditions : ITEMERR, LENGERR, QIDERR, NOSPACE, NOTAUTH, SYSIDERR, IOERR, INVREQ, ISCINVREQ

READQ TS

- Syntax :

```
EXEC CICS READQ TS
                    QUEUE(name)
                   {INTO(data-area) | SET(ptr-ref) }
                    LENGTH(data-value)
                   [NUMITEMS(data-area)]
                   [ITEM(data-area) | NEXT ]
END-EXEC.
```

Conditions : ITEMERR, LENGERR, QIDERR, NOTAUTH, SYSIDERR, IOERR, INVREQ, ISCINVREQ

DELETEQ TS

- Deletes all entries in the queue

- Syntax :

```
EXEC CICS DELETEQ TS
                    QUEUE(name)
END-EXEC.
```

Conditions: INVREQ, ISCINVREQ, NOTAUTH, QIDERR, SYSIDERR

Recap

- What are Queues?
- What are the types of queues?
- What is the difference between TDQ and TSQ?
- Is random access possible in TDQ?
- How is random access done in TSQ?
- Can you delete a single item in TSQ?

Lab Session

Write the record to TDQ

```
000001   IDENTIFICATION DIVISION.
000002   PROGRAM-ID. TTSG.
000003   ENVIRONMENT DIVISION.
000004   DATA DIVISION.
000005   WORKING-STORAGE SECTION.
000006        COPY EMAP.
000007   01 REC1.
000008      02 ENO PIC X(4).
000009      02 ENAME PIC X(15).
000010   77 WS-RESP PIC S9(8) COMP.
000011   77 MSG PIC X(25).
000012   PROCEDURE DIVISION.
000013   PARA1.
000014       MOVE LOW-VALUES TO REC1.
000015       MOVE LOW-VALUES TO EMPI, EMPO.
000016       EXEC CICS SEND
000017           MAP('EMP') MAPSET('MAPLE41')
000018       END-EXEC.
```

```
•   000019          EXEC CICS RECEIVE
•   000020              MAP('EMP') MAPSET('MAPLE41')
•   000021          END-EXEC.
•   000022          MOVE ENOI TO ENO.
•   000023          MOVE ENAMEI TO ENAME.
•   000024          EXEC CICS WRITEQ TD
•   000025              QUEUE('EQ41')
•   000026              FROM(REC1)
•   000027              RESP(WS-RESP)
•   000028          END-EXEC.
•   000029          IF (WS-RESP = DFHRESP(NORMAL))
•   000030              MOVE "DATA WRITTEN INTO Q" TO MSG
•   000031          ELSE
•   000032              MOVE "NOT WRITTEN TO Q" TO MSG.
•   000033          EXEC CICS SEND
•   000034               FROM(MSG)
•   000035               LENGTH(LENGTH OF MSG)
•   000036          END-EXEC.
•   000037          EXEC CICS RETURN END-EXEC.
•   000038          STOP RUN.
```

Read TDQ Queue and send the map

```
•  000001    IDENTIFICATION DIVISION.
•  000002    PROGRAM-ID. TTTK.
•  000003    ENVIRONMENT DIVISION.
•  000004    DATA DIVISION.
•  000005    WORKING-STORAGE SECTION.
•  000006        COPY EMAP.
•  000007    01 REC1.
•  000008      02 ENO PIC X(4).
•  000009      02 ENAME PIC X(15).
•  000010    77 WS-RESP PIC S9(8) COMP.
•  000011    77 MSG PIC X(25).
•
```

```
000012    PROCEDURE DIVISION.
000013    PARA1.
000014        EXEC CICS READQ TD
000015            QUEUE('EQ41')
000016            INTO(REC1)
000017        END-EXEC.
000018        MOVE ENO TO ENOO.
000019        MOVE ENAME TO ENAMEO.
000020        EXEC CICS SEND
000021            MAP('EMP') MAPSET('MAPLE41')
000022        END-EXEC.
000023        EXEC CICS RETURN END-EXEC.
000024        STOP RUN.
```

Write Records in TSQ

```
•   000001    IDENTIFICATION DIVISION.
•   000002    PROGRAM-ID. TTSG.
•   000003    ENVIRONMENT DIVISION.
•   000004    DATA DIVISION.
•   000005    WORKING-STORAGE SECTION.
•   000006           COPY EMAP.
•   000007    01 REC1.
•   000008       02 ENO PIC X(4).
•   000009       02 ENAME PIC X(15).
•   000010    77 WS-RESP PIC S9(8) COMP.
•   000011    77 MSG PIC X(25).
•   000012    PROCEDURE DIVISION.
•   000013    PARA1.
•   000014         MOVE LOW-VALUES TO REC1.
•   000015         MOVE LOW-VALUES TO EMPI, EMPO.
•   000016         EXEC CICS SEND
•   000017             MAP('EMP') MAPSET('MAPLE41')
•   000018         END-EXEC.
```

```
000019     EXEC CICS RECEIVE
000020         MAP('EMP') MAPSET('MAPLE41')
000021     END-EXEC.
000022     MOVE ENOI TO ENO.
000023     MOVE ENAMEI TO ENAME.
000024     EXEC CICS WRITEQ TS
000025         QUEUE('MP41MP41')
000026         FROM(REC1)
000027         RESP(WS-RESP)
000028     END-EXEC.
000029     IF (WS-RESP = DFHRESP(NORMAL))
000030         MOVE "DATA WRITTEN INTO Q" TO MSG
000031     ELSE
000032         MOVE "NOT WRITTEN TO Q" TO MSG.
000033     EXEC CICS SEND
000034          FROM(MSG)
000035          LENGTH(LENGTH OF MSG)
000036     END-EXEC.
000037     EXEC CICS RETURN END-EXEC.
000038     STOP RUN
```

Read the item from TSQ

```
•   000001    IDENTIFICATION DIVISION.
•   000002    PROGRAM-ID. TTTK.
•   000003    ENVIRONMENT DIVISION.
•   000004    DATA DIVISION.
•   000005    WORKING-STORAGE SECTION.
•   000006       COPY EMAP.
•   000007    01 REC1.
•   000008      02 ENO PIC X(4).
•   000009      02    PIC X.
•   000010      02 ENAME PIC X(15).
•   000011      02    PIC X(60).
•   000012    77 WS-RESP PIC S9(8) COMP.
•   000013    77 MSG PIC X(25).
•   000014    77 ITEMNO PIC S9(4) COMP.
•   000015    PROCEDURE DIVISION.
•   000016    PARA1.
•   000017       MOVE 2 TO ITEMNO.
```

```
000018       EXEC CICS READQ TS
000019         QUEUE('MP41MP41')
000020         INTO(REC1)
000021         ITEM(ITEMNO)
000022       END-EXEC.
000023       MOVE ENO TO ENOO.
000024       MOVE ENAME TO ENAMEO.
000025       EXEC CICS SEND
000026         MAP('EMP') MAPSET('MAPLE41')
000027       END-EXEC.
000028       EXEC CICS RETURN END-EXEC.
000029       STOP RUN.
```

Delete the TSQ

```
000001    IDENTIFICATION DIVISION.
000002    PROGRAM-ID. GGB.
000003    ENVIRONMENT DIVISION.
000004    DATA DIVISION.
000005    WORKING-STORAGE SECTION.
000006    77 WS-RESP PIC S9(8) COMP.
000007    77 MSG PIC X(25).
000008    PROCEDURE DIVISION.
000009    PARA1.
000010        EXEC CICS DELETEQ TS
000011           QUEUE('MP41MP41')
000012           RESP(WS-RESP)
000013        END-EXEC.
```

```
000014        IF WS-RESP = DFHRESP(NORMAL)
000015          MOVE "Q DELETED" TO MSG
000016        ELSE
000017          MOVE "Q NOT DELETED" TO MSG.
000018        EXEC CICS SEND
000019          FROM(MSG)
000020          LENGTH(LENGTH OF MSG)
000021        END-EXEC.
000022        EXEC CICS
000023           RETURN
000024        END-EXEC.
000025        STOP RUN.
```

Try Yourself!

- Write a program to receive the student details in the map and write to the TDQ(also to TSQ)
- Write a program to access a particular student’s detail at random in TSQ.