Crash Course - Anatomy of a Cobol CICS/DB2 application

This was written for those new to zOS application development. As a reference, the IBM sample 'MortgageApplication' (MortApp), in this repo, will be used to introduce basic concepts, terminology and application design with a focus on CICS/DB2. As a aid in further learning, links to external material have been included.

Foundational concepts

Mainframe programs are written mostly in Cobol. Others can be in Assembler, PLI and other language. Applications are a set of one or more programs. Each program specializes in some specific business feature. Applications and the data they process can be designed as interactive (online) or as batch processes.

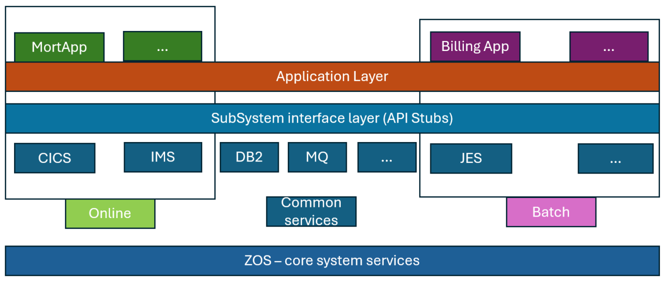
Interactive applications use the IBM product CICS or IMS.

- They are designed to interact with users to gather and send data over a network connected text-based 3270 terminal.
- Modern interactive applications can substitute 3270 screens with a Web front-end to access CICS programs and resources.

Batch applications run using Job Control Language - JCL.

- Batch is designed to process large amounts of data (batches) without user interaction.
- A JCL job is a sequence of step(s) that execute an application program or some utility like sort or DB2 bind.
- Steps also have Data Definitions (DDs) to allocate the files (Datasets) used by the program.
- Jobs are submitted to the Job Entry Subsystem - JES.

The diagram below illustrates the different layers of a mainframe application. zOS, the operating system is at the bottom and supervises all the work to support program execution and the hardware resources they use (not shown). Above zOS are the online and batch subsystems. Other subsystems, like DB2, are common across online and batch applications. The top layer represents mainframe applications. They access subsystem services through an API layer. For example, EPSCMORT accesses CICS services using 'EXEC CICS ...'. and it accesses DB2 resources with 'EXEC SQL ...'. These 'EXEC' statements are translated by the Cobol compiler to call subsystem stub programs to handle the data exchange.



Anatomy of a CICS Application

A basic CICS application has several parts. In our example, we will look at the source code used by DBB to build the MortApp.

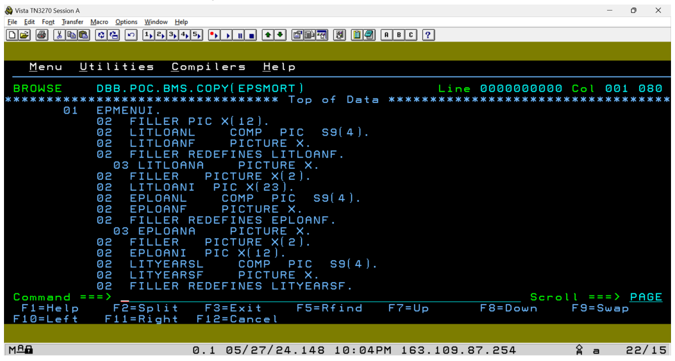
The code

MortApp is a basic online CICS application made up of several programs:

- cobol/eps**c**mort.cbl is the main program. It uses the "EXEC CICS" api to call **bms/epsmort**.

- bms/epsmort.bms is a 3270 screen BMS program written in assembler language.
  - The compiler transforms this source file into 2 artifacts; a symbolic copybook 'EPSMORT' and a physical executable load module.
  - The BMS copybook is saved to a Partitioned Dataset - PDS using the dbb-zappbuild "HLQ' arg.
  - This PDS is then used as a SYSLIB.
  - SYSLIB is a DDname to allocate a copybook PDS as input to the compiler. This is how the Cobol program's 'Copy' statement gets its copybook source during a build epscmort.

Example epsmort copybook member



```
   Menu  Utilities  Compilers  Help
 BROWSE    DBB.POC.BMS.COPY(EPSMORT)                   Line 0000000000 Col 001 080
********************************* Top of Data **********************************
       01  EPMENUI.
           02  FILLER PIC X(12).
           02  LITLOANL     COMP  PIC  S9(4).
           02  LITLOANF     PICTURE X.
           02  FILLER REDEFINES LITLOANF.
             03 LITLOANA     PICTURE X.
           02  FILLER   PICTURE X(2).
           02  LITLOANI  PIC X(23).
           02  EPLOANL     COMP  PIC  S9(4).
           02  EPLOANF     PICTURE X.
           02  FILLER REDEFINES EPLOANF.
             03 EPLOANA     PICTURE X.
           02  FILLER   PICTURE X(2).
           02  EPLOANI  PIC X(12).
           02  LITYEARSL     COMP  PIC  S9(4).
           02  LITYEARSF     PICTURE X.
           02  FILLER REDEFINES LITYEARSF.
 Command ===> _                                               Scroll ===> PAGE
  F1=Help     F2=Split    F3=Exit     F5=Rfind    F7=Up       F8=Down     F9=Swap
 F10=Left    F11=Right   F12=Cancel

M                      0.1 05/27/24.148 10:04PM 163.109.87.254            a    22/15
```

```
 A special note on DBB builds is that BMS copybooks are not stored in the source repo like other copybooks.  Instead they are stored in the
```

- cobol/epscsmrt.cbl is a program that is called by EPSCMORT to calculate a mortgage.
  - The data is returned using a COMM-AREA.
  - In Cobol, COMM-AREAs are data structures defined as copybooks within the 'Linkage Section' to exchange data.

- copybook/epsmtcom.cpy is the COMM-AREA copybook used between EPSCMORT and EPSCSMRT. This copybook includes 2 other copybooks for input and output data definitions.

The infrastructure

Once the MortApp is built, it needs to be defined to CICS and DB2. This section outlines the Jobs used for the definitions.

CICS Application Defintions

- Transactions are what drive CICS applications.
  - EPSP is the main MortApp **Transaction ID** (tranid). All CICS applications must have at least one tranid.
  - When this tranid is entered on a CICS 3270 screen, CICS starts program EPSCMORT.
  - The transaction also defines what group its in. A CICS group is a set resources that are common to an application. In this case, EPSMTM is the group name for the MortApp.

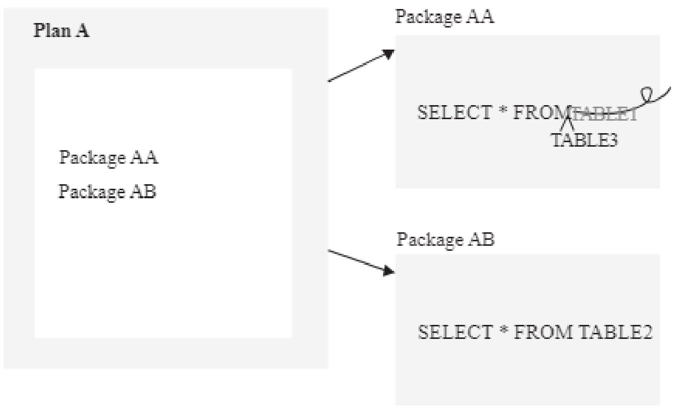
- Transactions and all other CICS resources are defined using the IBM batch utility DFHCSDUP , as in this example. Or they can be defined with the CICS tranid CEMT here is a list of other CICS utilities. In the example JCL you will see lines (control cards) that define:
  - DB2CONN - that is used to connect the MortApp to the DB2 subsystem 'DBD1'. It also defines 'EPSPLAN' as the DB2 plan for this group (see DB2 definitions below).
  - DB2ENTRY - defines DB2 properties related to all transaction for the EPSMTM group.
  - MAPSET - is the physical BMS load module that displays a 3270 screen (map)
  - PROGRAM - are the individual programs that make up the rest of the MortApp.

CICS System Layer

DB2 Application Definitions

As illustrated below, programs are defined to DB2 using a Plan. Plans are collections of DB2 packages. A package represents the DB2 resources used by a program.

When a DB2 program is compiled, a DB2 DBRM artifact is created. The DBRM is then bound to DB2 to update its resource requirement before it could run.



- epsbind.jcl binds the EPSCMORT package.
  - The conrtol cards for the bind utility follow "SYSTSIN DD *".
  - The DSN control card connects to the DBD1 DB2 subsystem.
  - The BIND package point to the PDS member EPSCMORT in the PDS alloacted with the "//DBRMLIB DD ..." card. Thats where DBB stored the DBRM when it built EPSCMORT.
  - Bind Package names the plan 'EPSPLAN' and includes the collection of packages for the application (PKLIST).
  - The Plan is defined once. But whenever a package is changed and a new DRBM creatd, it must be bind again.

DB2 System layer

RACF Security
RACF is the security subsystem for zOS. There are others like Top Secret and ACF2. In the MortApp, access to its DB2 resources is from CICS is defined in RACF

The racfdef.jcl job is run once to define the permission to access these CICS/DB2 resources:

- RDEFINE FACILITY DFHDB2.AUTHTYPE.DBD1 - defines the security profile for the **DB2CONN** CICS resource defined in DFHCSDUP job.
- RDEFINE FACILITY DFHDB2.AUTHTYPE.EPSE defines security for the **DB2ENTRY** resource.
- The 'PE' cards define which RACF User can access these profiles.
  - In a WaaS environment, the IBMUSER is a special (root-like) user that is given access.
  - The CICSUSER is CICS's RACF ID is also given access.
- The RDELETE cards are used to clean out the definitions when rerunning the job.

Other useful terms

... tbd

core

There are severla layers of mainframe programmering to consider

1. Operating System it selft. In this case we are using z/OS ver 3.1. That the latest IBM release as of 2024.

- System's programmers (sysProg) are responsibenl for installign and maintaint this layer -
- They also install key system products like CICS, DB2 and others. These are sometimes called subsystems.

2. Subsystem Admin - These are Adminstrators for each type of subsystems with epertise in designing, configuring, maintaining subsystems

- CICS Admins - work with Application developement teams to degien and support their applications
- Database Admin (DBA) - is responsible for simialr tasks for DB2
- There can be many other roles like the MQ Admin, RACF, Networking, Data storage

3. The application layer, This

There are other groups like those who manage the physical hardawre and the operations team who monitor day to day acitites and report issues to owning appliaciton or systems teams.

Anatomy of a CICS DB2 application:

- using the cobol programms in the MortApp repo, you will see folder for
  - bms - which is the source of Basic Mapping Support (bms or maps for short) files. I know I said MortApp is a Cobol program but maps are an exceptin. They are assmebler progrsm. There job is the 'map' text and input fields o na 3270 screen to get/set data to a user. bms/epsmort is the main bms map. NOrmally the main map is a menu whith options ot other maps (Screen).
  - cobol - is the where the main code lives. EPSCMORT is the main program that starts the application process.
  - copybook - in this example are files used by cobol prgrmas to define certain data structures. They are typcail created when 2 or more pgms needs share the same data layout - also called a record layout.

##So how does this all work?
A CICS program is 'installed' into a cics subsystem also called region. A region is just a Started Task (STC) running all the time like a deamon in the distributed world. An STC is like any JCL Job submitted to JES (Job Entry Subsystm) to allocate files (DDs) and run programs.

Some jobs, like batch applications, start and end in a short amount of time. Others, like CICS and DB2 run all the time.

Ther can be many parts to a CICS program. In this example, we will look at the runtime aspects like:
- load modules:
- Just like batch programs, pgms not executed in CICS, CICS programs are complied and linked and stored in a load PDS.
- In CICS the load pds is also called the Resource Parameter List or DFH**RPL** DD statement of the CICS STC JCL. Take a look ....
- changes to load modules will require a special CICS cmd to refresh it. Thats callaed a 'newcopy'
- If the program uses DB2, a Database Request Module (DBRM) is created as part of the compile. It must bound to the DB2 subsytem used by the applicaiotn.
- ...
-
like another other program. The diffenrebce is it will include some CICS system programs to , CICS and DB2 defiA CICS application has many parts To create a

.... wasteland

The also use several 'system' level services provided by CICS and in out case DB2.

When designing an applicaiotn, teams normally collaborate with zOS system admins to help archtect the optimum design that also follows some organizational standard.

program designed has several CICS resource definition; program, map, transaction and more.