# Design Challenge Notes

## Background

MQA (Medicare Quality Assurance) is highly modular, calling many utility subprograms each of which has its own unique record copybook(s). The MQA system receives claims from MQR (Medicare Quality Release), the CMS Claim Receipt and Control system, which reformats CWF (Common Working File) host claims from fixed to variable format, applies data-cleansing to all fields, synthesizes many fields, prepends a 49-byte header, and combines all host site daily transmittals into a single claim file.

MQA programs and subprograms perform COBOL business-rule edits to validate claim integrity, post error codes into the claims, collect summary error count metadata, modify certain claim fields, and generally prepare the claims for downstream data repositories. The claims are split into four file segments based on HICN ascending, and after editing are sorted into NCH (National Claims History) order, and reformatted to remove MQA-only edit data, before writing to database-digestible files.

## Objectives

The delivered solution must show that the prototype can handle the Medicare claim format and the MQA daily edits (as embodied in CMS demonstration COBOL program MQAPHHAB) by creating same *or similar* logic to ensure the following:

* Accept parms from JCL (or equivalent) to alter run characteristics: File date, ‘Test’ing S1/S\* to turn off/on DISPLAY (or similar) program process information capture
* Capture run metadata: Exercise new language equivalent of COBOL FUNCTION call
* Show successful claim I/O by reading, editing, and writing claims that can be value-mapped field to field to CMS production (e.g. MQAPHHAB) output

### Capture and display input and output file names for run reportage (similar to function of CMS ASSEMBLER module DDSDDCHK)

### Link to ‘external’ reference data: demonstrate ability to access external reference files which play in CMS production -

### In this case, invoke the MQANHCPC subprogram call for HCPCS code validation. MQANHCPC reads a linked-list file into working storage for recurring ultra-fast access. The file P@QAC.\*.CONTROL.LOAD is created from the CWF HCPC file by a proprietary CMS program which is not included in the demonstration requirements (MQAXHCPC)

### Perform or bypass Home Health Part A/B (HHA/B Split): Value Trailer XX, XY, XZ revenue line amount accumulation splits based on MQA RIC code (5) and NCH-NEARLINE-RIC-CODEs (U, W). NB: Though this is an ‘edit’, it changes claim data and thus cannot be bypassed. Correct execution of this logic is an important element in evaluating success

### Ability to handle some errors deliberately inserted into the sample dataset: error codes exist in MQAPHHAB but not triggered by dataset provided

* Produce report(s) showing file metadata and error counts in non-TEST mode, and program process notes, file metadata, and error counts/ranks in TEST mode, similar to MQAPHHAB

# Resources

CMS provide a sample dataset of 24 Home Health Claims for this prototype, drawn from production and scrubbed PII; and one test MAXCLAIM with all fields fully populated; a total of 25 claims.

DATA FILENAME: CMS Data File Name: B1D9.#MQR.OSDM.HHAB.D190729.EDTD

Design Challenge File Name: HHAB\_CLAIMS.BIN

HCPC FILENAME: CMS Mainframe File Name: [P@QAC.@BFA2701.CONTROL.HCPCS.LOAD.G0087V00](mailto:P@QAC.@BFA2701.CONTROL.HCPCS.LOAD.G0087V00)

Design Challenge File Name: HCPC\_LINKED\_LIST.BIN

MQAPHHAB TEST OUTPUT: CMS Test Output Report File: JH00\_EDTD\_DISP\_20190729

Design Challenge Sample Test Output Report File: MQAPHHAB\_TEST\_RUN.TXT

A list of the sample modules attached:

* MQAPHHAB:

I/O and edits –

* + Claims in and out – Handle Medicare claim physical layout, up to ten physical claims to compose one logical claim. Input file is HHAB\_CLAIMS.BIN.
  + Home Health A/B edit prep – Build the 450-element revenue trailer ‘pass’ area needed to do the rvnu edits. (In production, this is built in the I/O driver and passed to the edit modules via Linkage, along with the base claim.)
  + QAR5-COMMON-EDITS: base claim edits – should post 0014, A002, 8904, Y001 appropriately
  + EDIT-REVENUES: traverse revenue trailers to post 46#V, 61#H, 51#Z error(s)
  + Home Health A/B Split – detect which claims require that Part A, B, (and other) rvnu line totals need to be captured into VALU-TRLR XX, XY, XZ
  + Support accessing only rvnu line split candidates in date order, access by values from VALU-TRLRs 62 & 63, sum them, and post them into VALU-TRLR XX, XY, XZ
  + Error code posting, ranking, and capture into claims – Error order of detection does not match error severity rank, must sort by rank (internal table) and capture first 10 into the claim error array
* MQANHCPC:
  + Called module (or similar) to read (on first access), the HCPC file created by MQAXHCPC (this program is not part of the demo, but the file is); and load it into MQANHCPC for super-fast access on subsequent calls. Input linked list file is HCPC\_LINKED\_LIST.BIN.
* DDSDDCHK: a CMS ASSEMBLER utility to capture input/output filenames at runtime: DEMO to use language utility if exists

## Additional Notes

**Question**: (for Home Health Demo Readme) What is the definition of milieu here?

**Response**:

For the purposes of the exercise it is important for the program to provide *comparable* functionality (such as specifying date, input and output data file names and test mode/run override). It is not necessary to mimic mainframe parameters specific to the application.

“PARM='MQAHHAB,MQAPHHAB,PS040.S\*=20190729-99999999” *milieu* refers to the string “MQAHHAB MQAPHHAB, PS040, S1 (or S\*) which follow an MQA production standard to define (for displaying report sysouts at runtime):

**SYSTEM**: MQAHHAB

**PROGRAM**: MQAPHHAB

J**CL STEP**: PS040

**File Segment:** S1 – MQA claim files are split into 4 ‘segments’ by BENE HICN. All four are sometimes processing concurrently, using same programs, and their jcl is crafted to supply the correct segment to the filenames and parms. *Note File Segment is not part of this Design Challenge exercise.*

**RUN OVERRIDE**: S\* - ‘\*’ triggers program ‘test run’ behavior, to output detailed info tracing program logic path and subprogram ‘CALL’ results; a ‘test’ run against 25 claims is feasible. A ‘test’ run against 6 million claims is not feasible.

**Question**: (for Design Challenge Notes) Will CMS provide the input and output S3 buckets, or should those be provisioned as part of the solution?

**Response:**

The provided solution should provision the input and output S3 buckets. The solution should provide guidance to the reviewers on how to add files to the input bucket and how to retrieve files from the output bucket.