I wrote XML\_gen.pl based off the old version from Tim/Avry. It does XML generation ONLY (no upload/cli/deftable, those will be in a separate script).

It does do parallel job creation as well and matches jobs that are in parallel groups to their corresponding parallel job. I’d have to work out with Avry if this is the correct way to proceed.

Process:

1.       Set variables that will be used throughout the document. I changed how some variables were set and merged this into one method as well.

2.       Read and store configuration file into hashes. The methods called I didn’t change here, seems the algorithm is pretty straight forward. It stores up job names by sub table (key) and also creates a dependency map.

3.       Create SUB table XML’s.

a.       In previous iteration this was creating ctmgrpdef files.

4.       Create SUB table dependencies. This involves updating the SUB Table xml files with in/out condition elements

a.       In previous iteration they would appending the ctmgrpdef files with –OUTCOND flags

5.       Create JOB XML’s. For jobs in parallel groups it will create multiple files

a.       In previous iteration this was creating ctmdef files. Parallel jobs were handled after all generations were complete but keeping a placeholder in the job and memname.

6.       Create JOB dependencies. This will update XML’s with in/out elements. For jobs in parallel groups it matches parallel job to parallel job. If the successor is in a Group table it matches one to many or many to all type of relationship.

a.       In previous iteration they would appending the ctmgrpdef files with –OUTCOND flags. Parallels handled after job generation

7.       Merge files. First JOB XML’s are merged to their Sub-Tables. Then Sub-Tables are added to a SMART Table.

Here’s a run down of the files:

         Bill\_BillCycle\_C99M22Y2015.req is the input request file

         BillCycle\_cfg is the AMDOCS billing configuration file.

         It creates a folder C99\_M08\_Y2015\_R31\_P\_BLC that includes all XML files

o   XML files for SUB-Tables

o   XML files for JOBS

o   A merged XML noted below

         V21\_BLC\_C99R31 is the merged XML with SMART Table and all records

I am able to run **deftable** to write the table to EM successfully, upload to CTM and Order the jobs in my test lab. I ran the jobs in dummy mode and all jobs were linked and completed in proper order (eye balled but outlook so far so good)

Some things off the top that need to be improved/corrected:

1. Owner value should be dynamic. The table will need to include this value as well (for QA environments)
2. NodeID is set to a Nodegroup – we’d have to specify the nodegroup in Control-M OR get the actual hosts from OPHOST. I think this was Avry’s original intention.
3. Ensure job, group, app, and table names are all in line to what we agree to in the DDD.
4. Code readability – Some of the code could be cleaned up and/or made more efficient. A lot of the style and naming from the earlier scripts was retained
5. Max wait – 99? Automatically delete.
6. Confirm Group Name vs. Job Name - **FIXED – Original was using job name as the sub table name.**
7. Condition Excess