# Control-M Process for Creating/Ordering Billing Process

1. Deleting existing Control-M Table from EM database
   * If the table exists it will deleted, otherwise prints an error message. Table must be deleted to create a new Table.
2. Creating new billing run Table into EM database from XML file
3. Force upload changes from EM database to the Control-M database
   * Since the Table was not changed in Control-M, the statistics will be retained by existing jobs.
4. Add/Set Quantitative resource values
   * If the resource exists then set the resource. If request file specifies to hold the job then set resource to 0, otherwise set to 9999.
5. Order Control-M Table to Active Jobs.

## Workflow

Left side is detailed workflow. Right side is basic Control-M job setup. There will be the same setup for each Billing Type.



## Terminology

**DS job** -DS jobs are UNIX/Windows jobs managed by Control-M for Distributed Systems. One DS job represents one job converted from existing solution. Job names should follow existing convention from configuration files. Job streams are a collection of related DS jobs.

**SMART Table** - DS jobs are contained in a SMART table. A SMART table will represent one Billing Map. SMART table names should follow existing conventions and reflect the Billing Map.

**SUB-Table** -Sub containers of SMART tables. These can be used to identify groups of jobs that can run in parallel. For parallel ordering of jobs these would be ordered multiple times with different variables as required by the parallel run.

**Task Type** -Task type is always Command. Since Commands will be dynamic based on configuration parameters we will use variables to build commands on job order.

**Node** **ID** -Node ID is the server to execute job command. Node ID will be same as OPHOST value for existing solution; comparable to TWS CPU ID or WORKSTATION

**Owner** -Owner is the user ID to execute job command. This could be a static application account.

**AJF** – Active Job File, comparable to a TWS PLAN.

**Conditions** - Job flows often require dependencies between jobs so that some jobs run before or after other jobs. Control-M uses conditions (Out conditions and In conditions) to establish job dependencies. Most commonly, the same condition is defined as both:

**Out Condition** - an Out condition of a predecessor job. Out conditions are post-processing parameters; Control-M adds a job’s Out conditions to the active environment only when the job ends

**In Condition** – an In condition of a successor job. In conditions are job submission criteria; Control-M cannot submit a job until its In conditions have been added to the active environment (for example, by completion of the predecessor job).

When a predecessor job ends, the addition of its Out conditions to the active environment enables successor jobs with the corresponding In conditions to run

**Maybe Condition** – Certain jobs require In Conditions from predecessor jobs, but that predecessor job may not be scheduled on a particular day; the In Condition will still need to be satisfied by means of the CTMLDNRS utility. This utility will scan the AJF for matching IN and OUT Conditions and create a list of IN Conditions that do not match any OUT Conditions in the AJF. The utility can then be used to add the Condition by specifying the Condition name or a prefix and wildcard “\*”. Maybe conditions are typically prefixed with a character or characters. Macy’s will be using “M-“ and “MW-“ as Maybe Condition prefixes.

**New Day Procedure -** At the same time each day (known as New Day time), each Control-M/Server runs a procedure known as the New Day procedure. This procedure performs a number of tasks, including scheduling the day’s jobs, and running maintenance and cleanup utilities (for example, before adding the day’s jobs to the Active Jobs file, the New Day procedure deletes the old jobs from the previous day.)

**User Daily Job** - The New Day procedure can schedule all the current day’s jobs. However, it is more efficient to have the New Day procedure utilize a mechanism called User Daily jobs. User Daily jobs are job processing definitions whose sole purpose is to order jobs. Instead of directly scheduling production jobs, the New Day procedure can schedule User Daily jobs, and those User Daily jobs can schedule the production jobs.

**Order** - ordering the job or table to the AJF— an order request only schedules requested jobs whose scheduling parameters indicate that the jobs are eligible for scheduling that day.

**Force (order)** - forcing the job or table to the AJF— a force request schedules requested jobs regardless of whether their scheduling parameters are satisfied that day.

**RC -** Script Return Code after execution.

**ORDER ID** – a unique base 36 number (0-9, a-z) allocated to every Control-M AJF entity (Job or SMART/Sub Table). This number is used by Control-M to track each entity.

**Quantitative Resources** - User-defined variable representing a resource in the Control-M installation. The user defines the total quantity of this resource in the Control-M and, for each job, the quantity Required/used by that job. Control-M verifies that a job is not submitted for execution unless the Quantitative resources required by the job are available. Quantitative resources are recorded in the resources Table.

## Utilities Overview

This section outlines what utilities will be used and where they should be run from. Each Utility will have its own section on usage, output and exit code information.

### EM Utilities

These will all require a password file to run that specifies an EM user account (see below). User account requires FULL Table access to perform deletes/create and update operations.

1. (em) cli – Perform table operations such as Delete and Upload.
2. (emdef) deftable – Perform Table Definitions.

#### Password File

Store where the EM utilities will be located. It is flat text, unencrypted. Will be referenced as <passwordfile> in samples

user=<username>

password=<password>

### Control-M Utilities

These do not require a password file or credentials.

1. ecaqrtab – Add/Update Quantitative Resources
2. ctmorder – Orders the jobs into Control-M Active Jobs.

### Where to Run.

* EM utilities can run from anywhere an EM component is installed.
* Control-M utilities can run wherever a Control-M Agent is installed or on the Control-M Server host.
* If an agent is on the EM Server than all utilities can run from the same host.

# Utility Detail

## Table Delete (cli)

Example parameters as follows

em cli –pf <passwordfile> -h <em server> -TABLE\_DELETE <ctm\_server> <table\_name> Local

* <em\_server> is the EM server host
* <ctm\_server> is the Control-M host
* <table\_name> is the table to be deleted
* Local means the delete will only affect the EM database. The CTM database definition will not be updated.

### Samples

em cli -pf c:\rogers\passwordfile -h controlmdemo -TABLE\_DELETE controlmdemo smrt1 Local

#### Success Output – return code 1

Executed : -TABLE\_DELETE controlmdemo smrt1 Local

#### Failure Output – return code 1

Table was not found

Failed to execute -TABLE\_DELETE controlmdemo smrt1 Local

Reason [0] FAIL

## Table Create (deftable)

Example parameters as follows

emdef deftable –pf <passwordfile> -s <em\_server> -src <source\_file>

* <em\_server> is the EM server host
* <source\_file> is the Control-M host

### Samples

em cli -pf c:\rogers\passwordfile -s controlmdemo -src "C:\Rogers\scripts\table.xml"

#### Success Output – return code 0

1 out of 1 tables were successfully written to CONTROL-M/EM database.

#### Failure Output – return code 1

Error: Table controlmdemo BILLINGTYPE\_CYCLE\_RERUN already exists

### XML File

XML file must conform to specs in deftable.dtd. The included examples have minimum required fields and additional fields to meet our job requirements. All replaceable fields are prefixed \*\*. These will be used as basis or template for building a single XML file for loading.

#### *Attributes*

* APPLICATION and GROUP – Categorization of jobs for viewing/searching/reporting
* DATACENTER – Control-M Server Table belongs too
* TABLE\_NAME – Container for jobs in Control-M database. TABLE\_NAME and JOBNAME are used to track job statistics
* JOBNAME – Identify job with unique name. Describes Sub-Table name for Sub-Tables.
* MEMNAME – Identify jobs with unique name. It was discussed this will represent the current job name from existing solution (will include Month/Year component)
* TASKTYPE/CMDLINE – With TASKTYPE Command it mandates CMDLINE, which is the actual command the job is running.
* NODEID – Server from OP database where job will run
* OWNER – User account that runs the jobs
* DESCRIPTION – Describes the jobs
* CONFRIM – Set to 1 to put on user confirmation
* TIMEFROM – 24 hour clock for when jobs can start
* TIMETO – Set to > so jobs can continue past new day procedure (not needed if TIMEFROM is not set)
* MAXWAIT – set to 1 so waiting jobs will remain
* TABLE\_USER\_DAILY – Set to SYSTEM for automatic ordering. Default is empty (manual order only)

#### Elements

* RULE\_BASED\_CALENDAR – mandatory for SMART Tables. With only specifying the name the default setting is NO job schedule
* INCOND/OUTCOND – Match the names for creating dependencies. Conditions should be named Jobname-TO-Jobname to ensure unique conditions (Or Memname-TO-Memname).
* QUANTITATIVE – Throughput resource for billing jobs to suspend active jobs. Name will be same name as TABLE\_NAME for all jobs in a given Table. Value set to 1.
* CONTROL – Contention resource for daemon jobs. Name set to Job Name and type E for exclusive
* SHOUT – Used to send messages to the EM Alert monitor for job failure (WHEN=NOTOK) and for late execution (WHEN=EXECTIME)

#### Daemon Specific

* DAYS – Set to ALL for daemon jobs to run everyday
* JAN/FEB/MAR/APR/MAY/JUN/JUL/AUG/SEP/OCT/NOV/DEC – Set to 1 to allow schedule for these months. Default is 0 which is do not run.
* CYCLIC – Set to 1 to enable cyclic mode
* CYCIC\_TYPE/INTERVAL – Determines interval usage and sets rerun interval. “00005M” is every 5 minutes.
* IND\_CYCLIC – Determines when the job reruns. Set to END to be from job end time

## Table Upload (cli)

Example parameters as follows

em cli –pf <passwordfile> -h <em server> -TABLE\_UPLOAD <ctm\_server> <table\_name> -BY\_FORCE

* <em\_server> is the EM server host
* <ctm\_server> is the Control-M host
* <table\_name> is the table to be uploaded
* BY\_FORCE is required since the Control-M Server data is considered more to date than the newly created Table in EM database.

### Samples

em cli -pf c:\rogers\passwordfile -h controlmdemo -TABLE\_UPLOAD controlmdemo smrt1 –BY\_FORCE

#### Success Output – return code 1

Executed : -TABLE\_UPLOAD controlmdemo smrt1 -BY\_FORCE

Waiting for callbacks ...

Reply [to command: -TABLE\_UPLOAD controlmdemo *smrt1* -BY\_FORCE]

Request was accomplished...

#### Failure Output – return code 1

Failure would be in result Control-M being down or gateway disconnected.

Failed to execute -TABLE\_UPLOAD controlmdemo smrt1 -BY\_F

ORCE

Reason [30] NO CTM CONNECTION

In the even the table is not created in EM database (deftable fails).

Reason [20] OBJECT NOT FOUND

## Quantitative Resource (ecaqrtab)

Example parameters as follows

ecaqrtab ADD <QR name> <Max>

ecaqrtab UPDATE <QR name> <Max>

* <QR Name> is the resource name
* <Max> is the total availabile resources.
  + Add/Update to 0 for suspension of jobs
  + Add/Update to 9999 to jobs to be submitted.

### Samples - Adding

ecaqrtab ADD res1 5

#### Success Output – return code 0

Quantitative resource 'res1', Max availability '5' was added.

#### Failure Output – return code 1

Quantitative resource 'res1' already exists.

### Samples - Updating

ecaqrtab UPDATE res1 0

#### Success Output – return code 0

Quantitative resource 'res1', was updated from '5' to '0'.

#### Failure Output – return code 1

Quantitative resource 'res1' does not exists.

## Table Order (ctmorder)

Example parameters as follows

ctmorder –TABLE <table\_name> -NAME <job\_name> -ODATE <order date> -FORCE y

* <table\_name> is the Table name created in XML
* <job\_name> should be \* for all jobs in the table
* <order date> should be odat, to represent the same day as current day
* -FORCE y means to order the jobs regardless of schedule. Since no schedule will be defined this is necessary.

### Samples

ctmorder -TABLE smrt1 -NAME \* -ODATE odat -FORCE y

#### Success Output – return code 0

Performing FORCE of SMART Table 'smrt1' Job Name '\*' Odate '20150102'

5700 Table 'smrt1', Jobname '\*'

5701 SMART table 'smrt1' ordered, memname = '', orderno='0003oj'

5701 Sub table 'smrt1/sub1' ordered, memname = '', orderno='0003ok'

5701 JOB 'smrt1/sub1/testjob1' ordered, memname = '', orderno='0003ol'

5701 Sub table 'smrt1/sub2' ordered, memname = '', orderno='0003om'

5701 JOB 'smrt1/sub2/testjob3' ordered, memname = '', orderno='0003on'

#### Failure Output – return code 1

5703 TABLE smrT4 DOES NOT EXIST

# Job Naming Convention

Suggestions to be worked through with IBM/CFS/Rogers

### Daemons

|  |  |
| --- | --- |
| Table |  |
| Application |  |
| Group |  |
| Jobname | |  |  | | --- | --- | | Daemon Start |  | | Daemon Stop |  | | Daemon Check |  | |
|  |  |

### Billling – Launcher Jobs

|  |  |
| --- | --- |
| Table |  |
| Application |  |
| Group |  |
| Jobname | |  |  | | --- | --- | | Billing Type FW |  | | Validation |  | | OP Entries |  | | Trigger Job |  | |
|  |  |

### Billling Jobs

|  |  |
| --- | --- |
| Table | \*\*BILLINGTYPE\_\*\*CYCLE\_\*\*RERUN |
| Application | \*\*BILLINGTYPE\_\*\*CYCLE\_\*\*RERUN |
| Group | \*\*GROUP\_A|B|G |
| Jobname | \*\*GROUP\_B\*\*CYCLE{A|B|G}{\*\*PAR|00} |
| Memname | \*\*GROUP\_B\*\*CYCLE\*\*MM\*\*YYYY{A|B|G}{\*\*PAR|00} |