**Shared CC DB PATCHING INSTRUCTIONS**

**Pre-patch Validation Steps**:

1. Access Requirements:

* None as we assume DBA is running validation and has all requisite table accesses.
* Shared CC Production database url is uvacpmmora01mxd.maxcorp.maximus; port 1531

1. Check if all jobs have completed processing – run the following query to check.

**SELECT** *\** **FROM** cisco\_enterprise\_cc.cc\_a\_scheduled\_job **WHERE is\_running** = 1;

If the query returns no rows, then no jobs are running and we can proceed.

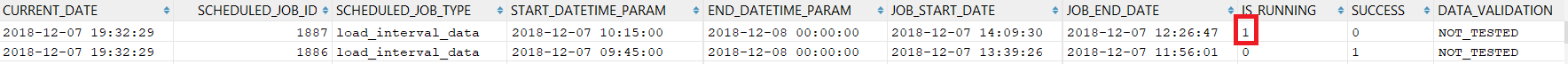
1. Work with the app admin to make sure all crons are turned off. The app admins are provided a pre-patching document with detailed steps on what to check/verify before turning the crons off.

**Post-patch validation Steps:**

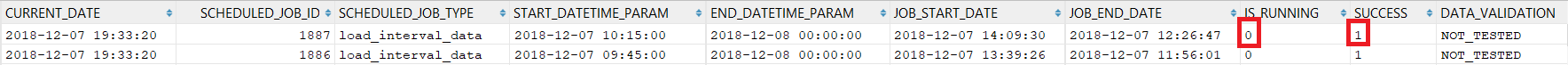
1. Work with app admin to ensure crons are turned back on after patching is completed.
2. After confirmation of crons being turned back on - Verify if the scheduled interval jobs (that run at the 10th and 40th min of the hour) are **running** and **completing** successfully as expected.

**SELECT** *CURRENT\_DATE*, **a**.*\** **FROM** CISCO\_ENTERPRISE\_CC.CC\_A\_SCHEDULED\_JOB **a WHERE SCHEDULED\_JOB\_TYPE** = **'load\_interval\_data' ORDER BY SCHEDULED\_JOB\_ID DESC**;

IS\_RUNNING should == 1 when the job is **running**.



After the job successfully completes, IS\_RUNNING == 0 and SUCCESS == 1

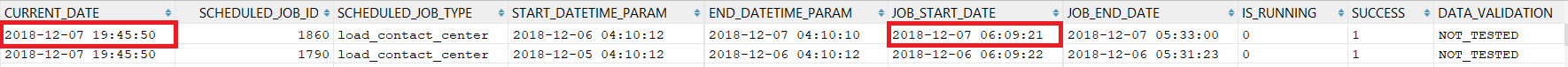


If load\_interval\_data entries aren’t being created every 30 minutes, then the cron jobs are not turned on and app admin needs to be notified.

If the load\_interval\_data fails with entries of IS\_SUCCESS = 0 and IS\_RUNNING = 0 (jobs have completed but not successfully) – this needs to be looked into for any database errors.

1. Verify if the scheduled contact center job (that runs once daily) is **ran that day** and **completed** successfully as expected – Please check if the date stamp matches to current date.

**SELECT** *CURRENT\_DATE*, **a**.*\** **FROM** CISCO\_ENTERPRISE\_CC.CC\_A\_SCHEDULED\_JOB **a WHERE SCHEDULED\_JOB\_TYPE** = **'load\_contact\_center' ORDER BY SCHEDULED\_JOB\_ID DESC**;



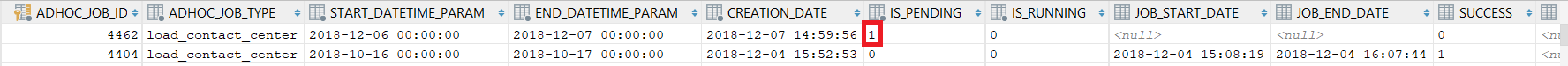
If the job was not run that day, then insert the job entry into adhoc table for adhoc run using **yesterday’s** date as **START\_DATETIME\_PARAM** and **today’s** date as **END\_DATETIME\_PARAM**.

For example, if today is **12/07/2018** and load\_contact\_center missed its run today, then the insert query will be

**INSERT INTO** CISCO\_ENTERPRISE\_CC.CC\_A\_ADHOC\_JOB(**adhoc\_job\_type**, **start\_datetime\_param**, **end\_datetime\_param**, **is\_pending**, **ACD\_SOURCE**, **WFM\_SOURCE**)  
**VALUES**(**'load\_contact\_center'**, **'2018-12-06 00:00:00'**,**'2018-12-07 00:00:00'**,1, **'CISCO'**, **'NA'**);  
**COMMIT**;

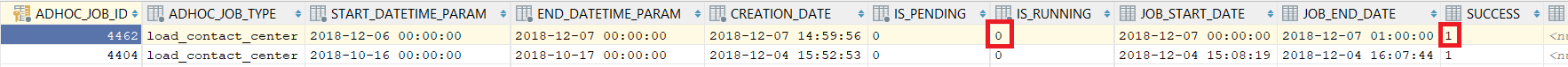
Check for the adhoc job to run (the job kicks off at the 35th minute of the hour):

**SELECT** *\** **FROM** CISCO\_ENTERPRISE\_CC.CC\_A\_ADHOC\_JOB **WHERE ADHOC\_JOB\_TYPE** = **'load\_contact\_center' ORDER BY ADHOC\_JOB\_ID DESC**;



Once the job kicks off, IS\_PENDING == 0 and IS\_RUNNING == 1

When the job finishes successfully the entry would be IS\_RUNNING == 0 and SUCCESS == 1



1. If the job is unsuccessful where IS\_RUNNING == 0 and SUCCESS == 0, please reach out to app admin to investigate logs. App admin has instructions on how to check adhoc logs for errors. If required to resolve the error, please reach out to available MAXDAT developer.
2. Add a new entry in scheduled table for the date we just ran, so the job doesn’t run again next day for the same day. If today is **12/07/2018**, the **START\_DATETIME\_PARAM** will be of yesterday’s date and **END\_DATETIME\_PARAM** will be of today’s date and **JOB\_START\_DATE** and **JOB\_END\_DATE** will come from the previous query’s **JOB\_START\_DATE** and **JOB\_END\_DATE**.

**INSERT INTO** CISCO\_ENTERPRISE\_CC.CC\_A\_SCHEDULED\_JOB (**scheduled\_job\_type**, **start\_datetime\_param**, **end\_datetime\_param**, **job\_start\_date**, **job\_end\_date**, **success**)  
**VALUES**(**'load\_contact\_center'**,**'2018-12-06 00:00:00'**,**'2018-12-07 00:00:00'**, *to\_date*(**'2018-12-07 00:00:00'**, **'YYYY-MM-DD HH24:MI:SS'**),   
 *to\_date*(**'2018-12-07 01:00:00'**, **'YYYY-MM-DD HH24:MI:SS'**),1);  
**COMMIT**;

1. Verify if the scheduled load\_ivr\_menu\_group job (that runs once daily) is **running** and **completing** successfully as expected.

**SELECT** *CURRENT\_DATE*, **a**.*\** **FROM** CISCO\_ENTERPRISE\_CC.CC\_A\_SCHEDULED\_JOB **a WHERE SCHEDULED\_JOB\_TYPE** = **'load\_ivr\_menu\_group' ORDER BY SCHEDULED\_JOB\_ID DESC**;



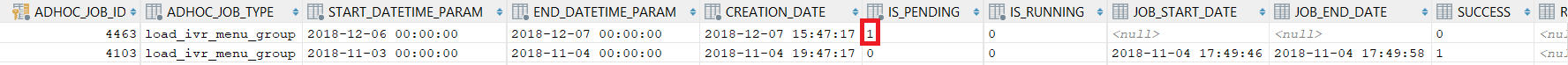
If the job ran successfully after patching, move on to step 10. If the job was scheduled to be run and is not showing up, then insert the job entry into adhoc table for adhoc run using **yesterday’s** date as **START\_DATETIME\_PARAM** and **today’s** date as **END\_DATETIME\_PARAM**.

For example, if today is **12/07/2018** and load\_ivr\_menu\_group missed its run today, then the insert query will be

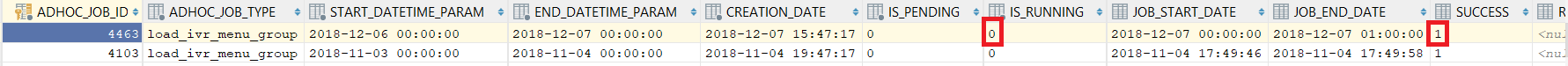
**INSERT INTO** CISCO\_ENTERPRISE\_CC.CC\_A\_ADHOC\_JOB(**adhoc\_job\_type**, **start\_datetime\_param**, **end\_datetime\_param**, **is\_pending**, **ACD\_SOURCE**, **WFM\_SOURCE**)  
**VALUES**(**'load\_ivr\_menu\_group'**, **'2018-12-06 00:00:00'**,**'2018-12-07 00:00:00'**,1, **'CISCO'**, **'NA'**);  
**COMMIT**;

Check for the adhoc job to run (the job runs at the 35th minute of the hour):

**SELECT** *\** **FROM** CISCO\_ENTERPRISE\_CC.CC\_A\_ADHOC\_JOB **WHERE ADHOC\_JOB\_TYPE** = **'load\_ivr\_menu\_group' ORDER BY ADHOC\_JOB\_ID DESC**;



Once the job kicks off, IS\_PENDING == 0 and IS\_RUNNING == 0. The job finished successfully when IS\_RUNNING == 0 and SUCCESS == 1.

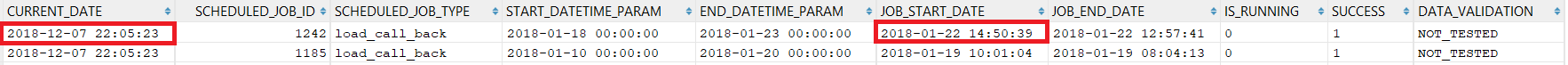


1. If the job is unsuccessful where IS\_RUNNING == 0 and SUCCESS == 0, please reach out to app admin to investigate logs. App admin has instructions on how to check adhoc logs for errors. If required to resolve the error, please reach out to available MAXDAT developer.
2. Add a new entry in scheduled table for the adhoc entry we just ran, so the job doesn’t run again next day for the same day. If today is **12/07/2018**, the **START\_DATETIME\_PARAM** will be of yesterday’s date and **END\_DATETIME\_PARAM** will be of today’s date and **JOB\_START\_DATE** and **JOB\_END\_DATE** will come from the previous query’s **JOB\_START\_DATE** and **JOB\_END\_DATE**.

**INSERT INTO** CISCO\_ENTERPRISE\_CC.CC\_A\_SCHEDULED\_JOB (**scheduled\_job\_type**, **start\_datetime\_param**, **end\_datetime\_param**, **job\_start\_date**, **job\_end\_date**, **success**)  
**VALUES**(**'load\_ivr\_menu\_group'**,**'2018-12-06 00:00:00'**,**'2018-12-07 00:00:00'**, *to\_date*(**'2018-12-07 00:00:00'**, **'YYYY-MM-DD HH24:MI:SS'**),   
 *to\_date*(**'2018-12-07 01:00:00'**, **'YYYY-MM-DD HH24:MI:SS'**),1);  
**COMMIT**;

1. Verify if the scheduled load\_call\_back job (that runs once daily) is **running** and **completing** successfully as expected.

**SELECT** *CURRENT\_DATE*, **a**.*\** **FROM** CISCO\_ENTERPRISE\_CC.CC\_A\_SCHEDULED\_JOB **a WHERE SCHEDULED\_JOB\_TYPE** = **'load\_call\_back' ORDER BY SCHEDULED\_JOB\_ID DESC**;



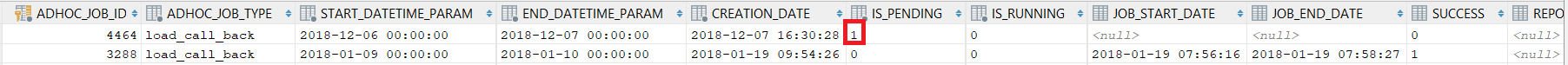
If the job ran successfully after patching, move on to step 12. If the job was scheduled to be run and is not showing up, then insert the job entry into adhoc table for adhoc run using **yesterday’s** date as **START\_DATETIME\_PARAM** and **today’s** date as **END\_DATETIME\_PARAM**.

For example, if today is **12/07/2018** and load\_call\_back missed its run today, then the insert query will be

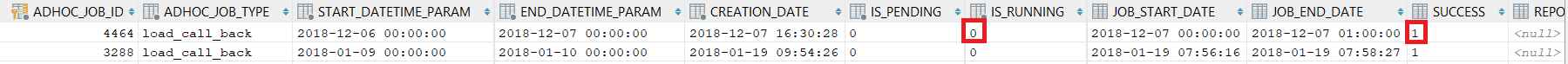
**INSERT INTO** CISCO\_ENTERPRISE\_CC.CC\_A\_ADHOC\_JOB(**adhoc\_job\_type**, **start\_datetime\_param**, **end\_datetime\_param**, **is\_pending**, **ACD\_SOURCE**, **WFM\_SOURCE**)  
**VALUES**(**'load\_call\_back'**, **'2018-12-06 00:00:00'**,**'2018-12-07 00:00:00'**,1, **'CISCO'**, **'NA'**);  
**COMMIT**;

Check for the adhoc job to run (the job runs at the 35th minute of the hour):

**SELECT** *\** **FROM** CISCO\_ENTERPRISE\_CC.CC\_A\_ADHOC\_JOB **WHERE ADHOC\_JOB\_TYPE** = **'load\_call\_back' ORDER BY ADHOC\_JOB\_ID DESC**;



Once the job kicks off, IS\_PENDING == 0 and IS\_RUNNING == 0. The job finished successfully when IS\_RUNNING == 0 and SUCCESS == 1.



1. If the job is unsuccessful where IS\_RUNNING == 0 and SUCCESS == 0, please reach out to app admin to investigate logs. App admin has instructions on how to check adhoc logs for errors. If required to resolve the error, please reach out to available MAXDAT developer.
2. Add a new entry in scheduled table for the adhoc entry we just ran, so the job doesn’t run again next day for the same day. If today is **12/07/2018**, the **START\_DATETIME\_PARAM** will be of yesterday’s date and **END\_DATETIME\_PARAM** will be of today’s date and **JOB\_START\_DATE** and **JOB\_END\_DATE** will come from the previous query’s **JOB\_START\_DATE** and **JOB\_END\_DATE**.

**INSERT INTO** CISCO\_ENTERPRISE\_CC.CC\_A\_SCHEDULED\_JOB (**scheduled\_job\_type**, **start\_datetime\_param**, **end\_datetime\_param**, **job\_start\_date**, **job\_end\_date**, **success**)  
**VALUES**(**'load\_call\_back'**,**'2018-12-06 00:00:00'**,**'2018-12-07 00:00:00'**, *to\_date*(**'2018-12-07 00:00:00'**, **'YYYY-MM-DD HH24:MI:SS'**),   
 *to\_date*(**'2018-12-07 01:00:00'**, **'YYYY-MM-DD HH24:MI:SS'**),1);  
**COMMIT**;

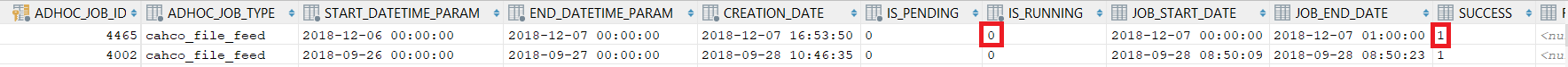
**CA HCO FILE FEED MISSING**

1. If the app admin reaches out regarding missing CA HCO files, we need to insert an adhoc entry in the adhoc table with **yesterday’s** date as **START\_DATETIME\_PARAM** and today’s date as **END\_DATETIME\_PARAM** parameters. If today is **12/07/2018**, the query is:

**INSERT INTO** CISCO\_ENTERPRISE\_CC.CC\_A\_ADHOC\_JOB(**adhoc\_job\_type**, **start\_datetime\_param**, **end\_datetime\_param**, **is\_pending**, **ACD\_SOURCE**, **WFM\_SOURCE**)  
**VALUES**(**'load\_call\_back'**, **'2018-12-06 00:00:00'**,**'2018-12-07 00:00:00'**,1, **'CISCO'**, **'NA'**);  
**COMMIT**;

1. Verify if the adhoc job (that runs at the 35th min of the hour) is completing successfully as expected.

**SELECT** *\** **FROM** CISCO\_ENTERPRISE\_CC.CC\_A\_ADHOC\_JOB **WHERE ADHOC\_JOB\_TYPE** = **'cahco\_file\_feed' ORDER BY ADHOC\_JOB\_ID DESC**;



1. Once the query has 1 in SUCCESS column, please let the app admin know to check for existence of new files.

**Updates to CC\_HCO\_F\_V2\_CALL**

This update is necessary to get the CC\_HCO\_F\_V2\_CALL table in sync with CC\_F\_V2\_CALL\_SV

1. Update CC\_A\_LIST\_LKUP table with the date on which load\_contact\_center job was run – usually sysdate. Please make sure the session is set to ‘DD-MON-YY’ format.

**alter session set nls\_date\_format = 'DD-MON-YY';**

**update cc\_a\_list\_lkup**

**set out\_var = sysdate**

**where name = 'CAHCO\_V2\_CALL\_UPDATE\_DATE'**

**and list\_type = 'CAHCO\_V2\_CALL\_UPDATE\_DATE'**

**and value = 'CA HCO';**

**commit;**

1. Execute the following procedure to update the CC\_HCO\_F\_V2\_CALL table

**exec MAXDAT\_ADMIN.UPDATE\_HCO\_V2\_CALL;**

Once the procedure executes successfully, execute the queries below to validate the table data matches to the semantic view.

**select trunc(a.datetime), count(\*) from cc\_f\_v2\_call\_sv a, cc\_d\_project b**

**where a.d\_project\_id = b.project\_id**

**and b.project\_name = 'CA HCO'**

**and trunc(a.datetime) >= '01-APR-19'**

**group by trunc(a.datetime)**

**order by trunc(a.datetime) desc;**

**select trunc(a.datetime), count(\*) from cc\_hco\_f\_v2\_Call a**

**where trunc(a.datetime) >= '01-APR-19'**

**group by trunc(a.datetime)**

**order by trunc(a.datetime) desc;**

Both the above queries should produce the same result set.