InfoSphere CDC –
How to Perform Initial Table Synchronization
# TABLE OF CONTENTS

**TABLE OF CONTENTS** .................................................................................................................................................. 2  
**INTRODUCTION** ................................................................................................................................................... 3 

**TABLE SYNCHRONIZATION METHODS** ............................................................................................................... 4 
  InfoSphere CDC Refresh Operation ......................................................................................................................... 4 
  Database Utilities ...................................................................................................................................................... 5 
  ETL Tools (e.g. DataStage)........................................................................................................................................ 6 
  *External Table Synchronization Procedure for CDC 6.5.2 and below* ................................................................. 6 
  *External Table Synchronization Procedure for IIDR 10.2 onwards* ................................................................. 10 

**STRATEGIES USED TO HANDLE LARGE TABLES** .............................................................................................. 12 
  Use Alternative Synchronization Method .................................................................................................................. 12 
  Parallelize tables in different subscriptions ........................................................................................................ 12 
  Refresh the table in smaller batches ...................................................................................................................... 12 
  What if database logs cannot be retained for the duration of the Refresh? ......................................................... 12
INTRODUCTION

There are various reasons when table synchronization between the source and target tables is required. InfoSphere CDC has a built-in Refresh function that also performs table synchronization but there may also be other external methods that are preferred. This document will discuss different options and considerations around table synchronization and InfoSphere CDC.

Table synchronization may be performed when:

(i) Populating an empty target table with the initial source image before the table is mirrored by InfoSphere CDC
(ii) The source and target table have become out of sync
(iii) A table was previously parked for a period of time
(iv) A DDL operation has changed the structure of the table.
InfoSphere CDC Refresh Operation

The InfoSphere CDC product itself has a Refresh operation that is used to perform a snapshot copy of the data for the specified tables. When you first configure a table for replication it will automatically be flagged for Refresh to be performed to synchronize the tables.

What are the Sequence of Events during a Refresh Operation?

(i) Truncate or clear the target table in order to bring a fresh copy of the source image (default behaviour)
   a. The user can disable to specify alternative logic for this event
   b. If the table mapping is configured with Live Audit an initial truncation is not performed to retain the audit trail.
   c. Subset refresh allows the user to specify a where clause to indicate the subset of records deleted at the beginning of a Refresh operation
   d. Differential refresh will only synchronize the target table with the current contents of the source table by applying only the differences. Therefore a truncate or clear is not performed.

(ii) Drop target table constraints and indices prior to refreshing tables
   a. The system parameter `refresh_loader_drop_index` can be configured to specify this behaviour. The parameter applies to DB2 LUW, SQL Server and Sybase platforms only.
   b. Query the source table, directly accessing the database
   c. The query on the source database will cause a table scan to provide the rows. CDC does not use an order by clause. Some databases such as Oracle allow a snapshot image whereas others do not resulting in an indoubt period
   d. If the database supports a native bulk load method of refreshing the data, this will be the faster, preferred method. However, there may be some limitations when this method cannot be used which will be highlighted below.

(iii) In each subscription Refresh is performed on the tables one at a time. By default the order is alphabetical by name but can also be configured.

What are the different Refresh methods used?

If the native database has a fast load method InfoSphere CDC will utilize the fast loader for optimized refresh performance. The supported platforms include DB2 LUW, Microsoft SQL Server, Teradata, Oracle, Sybase and Netezza. There are exceptions when the fast load cannot be used.

- The system parameter `refresh_allow_fast_loader` is explicitly set to false
- The tables contain Large Object Types (LOBs)
- The system parameter `refresh_end_on_error` is set to false
- Truncate operation is set to do not delete
- Table is mapped with adaptive apply or live audit mapping
- Partial refresh of table
- Table contains triggers
- Row level user exits are enabled
• Lack of permissions to fast load directory

If the fast load method cannot be used InfoSphere CDC will use JDBC apply.

**How to handle tables with referential integrity?**

The Refresh Order configured in the Management Console can be used to organize the order of tables to refresh to keep within the constraints imposed on the tables. The activity on the source tables should be quiesced during the refresh period. By default the table order is arranged alphabetically by SCHEMA then TABLE NAME. The refresh order should be in logical order parent first, then child.

**Database Utilities**

Different database platforms may have native utilities that allow copying data over into the same database platform. This would be an ideal option for homogeneous replication. Some examples include:

- Oracle has export/import functions and the Oracle Data Pump utility that allows you to create a snapshot of the data and copy it into another database. The Wiki contains a detailed procedure how this is done.

- Insert into select * from using remote database linked table

Some of the migration tools also have the capability to perform heterogeneous copies:

- SAP R3load tool commonly used for migrations
  - The Wiki contains a document that describes SAP migrations using this tool in more detail
IBM's Xenobridge tool is also a powerful tool used for database migrations
  - The Migration Factory team is very experienced using this tool in large scale migrations

ETL Tools (e.g. DataStage)
Using an ETL tool such as IBM’s InfoSphere DataStage to perform the initial synchronization is ideal for large tables that may take days to Refresh using the InfoSphere CDC product. InfoSphere DataStage is very fast in comparison and efficient in synchronizing the tables because of the high parallelism within the tool. However, we need to understand how to transition from the initial ETL load to normal InfoSphere CDC ‘mirroring’. Some of the considerations include operations that occur during the ETL load and also determining the bookmark or log position to start from once the synchronization completes. In the case of InfoSphere DataStage, the source table will be locked during the synchronization therefore no changes will be made to the source tables during this time.

The simplified version of the procedure can be used if there are no DML or key column changes made to the source tables during the synchronization period.

External Table Synchronization Procedure for CDC 6.5.2 and below

Simplified Procedure:
1. Create the subscription with Standard Apply and add all applicable table mappings
2. Set the replication method as Mirroring
3. Set the table status to Active. This is equivalent to a Mark Capture Point in the Management Console.

Management Console:
Command Line:

dmmarktablecapturepoint -I <instance name> -s <subscription list> [-A for all tables or -t <list of tables>]

4. Perform the synchronization via the InfoSphere DataStage ETL job.

5. The procedure is simplified if the tables to be synchronized have no DML transactions occurring on them during the load. If this is the case perform another mark capture point as in Step 3 to reset the bookmark to the current log position. Otherwise, proceed to Step 6.

6. If the application is not quiesced during the synchronization there are a few ways to mimic the Refresh While Active behaviour by configuring the product to handle the in doubt period.
   a. Temporarily configure the mirror_end_on_error system parameter to ignore apply errors

   ![Modify System Parameter]

   b. Temporarily enable Conflict Detection for all columns with Resolution Method ‘Source Wins’.

   ![Conflict Detection and Conflict Resolution Configurations]
7. Start Mirroring with Scheduled End to ‘Now’. Alternatively if the log position or log time is known when the batch job completed this can be specified.

Management Console:

Command Line:

```
dmstartmirror -I <instance name> -n -s <list of subscriptions> [-p <a specified log position> or -t <a specified log time>]
```

8. When the subscription has caught up, replication will stop
   a. Reset the `mirror_end_on_error` system parameter to true
   b. Deselect the conflict detection for all columns
10. Restart mirroring in Continuous mode

   `dmstartmirror -I <instance name> -n -s <list of subscriptions> -c`

Procedure (If DML operations result in key column changes):

For new subscription:
1. a) Create the subscription with Adaptive Apply and add all applicable table mappings

For existing subscription:
1. b) Remap Source Table and Specify Adaptive Apply mapping. Please note that target mapping configurations are not saved and will need to be reset. This includes target derived expressions, row level user exits, value translations, key definitions, conflict detection resolution settings, MBCS configurations, column mappings, etc.
2. Perform the synchronization via the InfoSphere DataStage ETL job.
3. Start Mirroring with Scheduled End to ‘Now’. Alternatively if the log position or log time is known when the batch job completed this can be specified.

Management Console:

![Start Mirroring](image)

Command Line:

```
dmstartmirror -I <instance name> -n -s <list of subscriptions> [ -p <a specified log position> or -t <a specified log time>]```

4. When the subscription has caught up, replication will stop
5. Remap the subscription to Standard Replication similar to step 1b
6. Restart mirroring in Continuous mode

```
dmstartmirror -I <instance name> -n -s <list of subscriptions> -c```
External Table Synchronization Procedure for IIDR 10.2 onwards

With the release of IIDR 10.2 two new commands have been added to allow an easier and automatable way of integrating external table synchronizations with CDC. The commands `dmmarkexternalunloadstart` and `dmmarkexternalunloadend` allow the user to specify the beginning and ending positions of the log that mark the in-doubt period. Afterwards, when mirroring is initiated the changes that occurred during the external table synch will be reconciled by running CDC in an adaptive apply mode ignoring conflicts that may have occurred during the in-doubt interval. These commands are not applicable to CDC for IBM i.

**Procedure for Linux, Unix, Windows based platforms:**

1. Stop the subscription if it is currently running
2. Invoke the command-line to mark the starting position:

   ```
dmmarkexternalunloadstart -I <instance name> -s <subscription> -t <table name>
   ``

   Successfully marked the unload start point

   CDC will also issue a mark capture point at this time.

3. Use external tools to unload the table data (see limitations on transaction isolation level used by the tools)
4. After unload completes, invoke command-line to mark the end position:

   ```
dmmarkexternalunloadend -I <instance name> -s <subscription> -t <table name>
   ``

   Successfully marked the unload stop point
5. Use external tools to load table data to the target
6. Start subscription. CDC will reconcile the differences corresponding to the changes made to the source table during the synchronization phase. CDC runs in an Adaptive Apply manner during the range marked by the start and end commands.

Procedure for z/OS platforms:
1. Stop the subscription if it is currently running
2. Perform table synchronization outside of product, noting log positions at start and end of source table capture
3. Issue command
   SETMIRSTS,...,STATUS=REFRESH, LOGRANGE=(StartPosition,EndPosition)
4. Start mirroring

Transaction mode compatibility for unload operation:

<table>
<thead>
<tr>
<th>Database</th>
<th>True EOP</th>
<th>Partial TXN</th>
<th>Uncommitted</th>
<th>Committed</th>
<th>Repeatable</th>
<th>Serializable</th>
<th>Snapshot</th>
<th>Readpast</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2</td>
<td>NO</td>
<td>Process</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Oracle</td>
<td>YES</td>
<td>Discard</td>
<td>N/A</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Oracle Trigger</td>
<td>YES</td>
<td>Undefined</td>
<td>N/A</td>
<td>YES</td>
<td>N/A</td>
<td>YES</td>
<td>YES</td>
<td>N/A</td>
</tr>
<tr>
<td>MSSQL</td>
<td>No</td>
<td>Process</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Sybase</td>
<td>YES</td>
<td>Undefined</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>N/A</td>
<td>NO</td>
<td>N/A</td>
</tr>
<tr>
<td>Informix</td>
<td>YES</td>
<td>Undefined</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Solid</td>
<td>NO</td>
<td>Process</td>
<td>N/A</td>
<td>YES</td>
<td>YES</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>
STRATEGIES USED TO HANDLE LARGE TABLES

Use Alternative Synchronization Method

If the product Refresh operation takes a long time to complete consider using one of the external methods described above to synchronize the tables outside of the product and then following the procedures to set the appropriate mirroring point.

Parallelize tables in different subscriptions

The tables in each subscription will be refreshed one at a time in the default sequence of refresh order sequence before transitioning into mirroring mode. To allow refreshes to run in parallel one can consider splitting the tables into additional subscriptions to run the refresh operations together. If you decide to use this method, keep in mind that more subscriptions introduce more threads and resource utilization. It is not ideal to have too many subscriptions running when mirroring therefore you can consolidate the tables back into fewer subscriptions after the refreshes complete.

Refresh the table in smaller batches

The InfoSphere CDC Subset Refresh can be useful to address table synchronization concerns for large tables. Subset Refresh can be used to address the following cases:

1) Re-synchronizing a known portion of the tables which are out of synch
   By default the Refresh operation will truncate and refresh the entire table. In the event that a known subset of the table has become out of synch we have the option of specifying a where clause that indicates the subset of data to delete from the target and select from the source. For example, specify a subset by product id, date range, or region id.

2) Refreshing large tables in stages
   a) Accommodate smaller batch windows
   b) Less interruption for other tables being replicated

3) Synchronization check for subset of rows
   The Subset Refresh can be used in conjunction with the Differential Refresh feature to mimic the behaviour of a synch check. Since Differential Refresh performs a table scan of the table, this check would only be suitable for a manageable number of rows.

The differential and subset refresh cannot use the fast load method but uses JDBC instead. For more information about Subset Refresh and Differential Refresh see the presentation IS_CDC_65_Partial_Refresh on the CoE Wiki

What if database logs cannot be retained for the duration of the Refresh?

It is best practice to use the InfoSphere CDC dmshowlogdependency command as part of the customer’s log retention cleanup script. In the event that the logs cannot be retained long enough to wait for the refresh to complete and catch up, one option is to enable Continuous Capture. If Continuous Capture is used additional disk space must be accounted for to store the changes that are performed during the long running refresh.
Disclaimer: By enabling the Continuous Capture feature additional disk space would be required to account for the changed data when the subscription is not running. Performance will also be impacted because the product will read the changes from disk instead of memory. This procedure should only be considered as a temporary solution if a minimal set of logs cannot be retained before CDC has finished reading from them. The feature should not be used for regular business processing.

1. Enable Continuous Capture
   
   `dmenablecontinuouscapture -I <instance name>`

2. The continuous capture feature will scrape changes only if there is at least one ‘active’ table. An active table means that the replication status of the table is ‘Mirror-Active’ (the subscription does not need to be running). If you do not have any other tables in another subscription set as active then you may need to create a dummy subscription and set a table to Mirror-Active. This will trigger the continuous capture to start.

3. Flag the subscription for refresh

4. Start Mirroring the subscription. Since the table is marked for Refresh, a refresh will kick off first and complete before transitioning to mirroring mode. Even if the refresh takes longer than the log retention period, when it commences mirroring CDC will have processed the operations in the logs that have since been purged and stored them in the CDC staging store.

5. After CDC has caught up you can disable the Continuous Capture feature
   
   `dmdisablecontinuouscapture -I <instance name>`