**Change Data Capture (CDC) Implementation Procedure**

**Prerequisites:** Make sure SQL Server Agent Service is running in order to get notified of the CDC actions

**Steps**:

1. Create the Database, if not exists

USE master

GO

IF NOT EXISTS(SELECT 1 FROM sys.databases WHERE name = 'TestDb')

CREATE DATABASE TestDb

GO

1. Enable CDC for the database

USE TestDb

GO

EXEC sys.sp\_cdc\_enable\_db

GO

**Note**: If you try to enable CDC for a database, and CDC is already enabled, then you see the following error message - Database 'TestDb' is already enabled for Change Data Capture. Ensure that the correct database context is set, and retry the operation. To report on the databases enabled for Change Data Capture, query the is\_cdc\_enabled column in the sys.databases catalog view.

Once CDC has been enabled for a database, a new column **is\_cdc\_enabled** is added to sys.databases table.

To check whether the CDC has been enabled for a database:

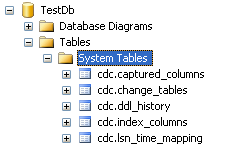
SELECT is\_cdc\_enabled

FROM sys.databases

WHERE name = 'TestDb'

It should return 1, if enabled; 0 otherwise.

Besides this, a new schema named ‘cdc’ and a new user named ‘cdc’ will be created. Also, following tables will be created. (related to CDC). These tables are populated when the first table is enabled for CDC.



1. Create a table, if not exists

CREATE TABLE Employee

(

EmployeeID INTEGER IDENTITY (1,1)

, EmpName VARCHAR(100)

, EmpCity VARCHAR(100)

CONSTRAINT [PK\_Employee\_EmployeeID] PRIMARY KEY CLUSTERED (EmployeeID ASC)

)

GO

1. Enable CDC for the table.

EXEC sys.sp\_cdc\_enable\_table

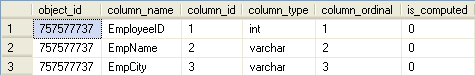
@source\_schema = 'dbo'

, @source\_name = 'Employee'

, @role\_name = 'cdc\_test' –A new role created, if not exists; and the user 'cdc' has been assigned to this role

The contents of the abovementioned tables are as below (after enabling CDC for Employee table) –

SELECT \* FROM cdc.captured\_columns



**Note (1)**: If a table is CDC enabled, and If you make any schema changes to the table, you need to disable CDC at the table level and then enable again in order to capture the data changes for the new columns

USE TestDb

GO

ALTER TABLE Employee ADD EmpSalary money

GO

To disable CDC on a table -

EXEC sys.sp\_cdc\_disable\_table

, @source\_schema = 'dbo'

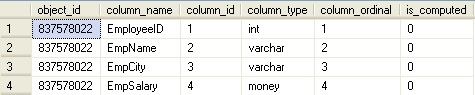
, @source\_name = 'Employee'

, @capture\_instance = 'dbo\_Employee'

, @supports\_net\_changes = 1

--(or 'all', if you have more than one capture\_instance name)

If you query the cdc.captured\_columns table again, it looks like this:



SELECT \* FROM cdc.change\_tables





SELECT \* FROM cdc.ddl\_history (Donot query this table directly, rather use the stored procedure as shown below)

EXEC sys.sp\_cdc\_get\_ddl\_history @capture\_instance = 'dbo\_Employee'



SELECT \* FROM cdc.index\_columns



SELECT \* FROM cdc.lsn\_time\_mapping

This table contains all the audit happened for the table, mapping Log Sequence Number (LSN) with the time of transaction

**Note (2)**: When the first table in a database has been enabled for change data capture, two new jobs are created and started by SQL Server Agent, as mentioned below -

Job 'cdc.TestDb\_capture' started successfully.

Job 'cdc.TestDb\_cleanup' started successfully.

**Note (3):** When a table has CDC enabled, a new change table is created which corresponds to the main table. When the name of the change table is not specified at the time the source table is enabled, the table is named as: **cdc.*capture\_instance*\_CT** where *capture\_instance* is the combination of schema name of the source table and the source table name in the format *schema\_table.* (Ex: **cdc.dbo\_Employee\_CT**).

This table contains all the audit information that happened for the main table.

**Note (4):** When a table has CDC enabled, 2 new fucntions are created

* fn\_cdc\_get\_all\_changes\_ dbo\_Employee\_CT
* fn\_cdc\_get\_net\_changes \_dbo\_Employee\_CT (**Note**: this function will be created if @supports\_net\_changes = 1 while enabling CDC for the table)

**Testing Change Data Capture**:

Perfrom some Data Modifications (Inserts, Updates & Deletes) on the main table. You can see that these opetations are entered into the change table automatically.

**Querying the change data from the change table:**

DECLARE @begin\_time datetime

DECLARE @end\_time datetime

DECLARE @from\_lsn binary(10)

DECLARE @to\_lsn binary(10);

declare @rowfilter varchar(10)

SET @rowfilter = 'all'

SET @begin\_time = DATEADD(hh,-1,GETDATE()); --going 1 hr back

SET @end\_time = GETDATE();

SELECT @from\_lsn = sys.fn\_cdc\_map\_time\_to\_lsn('smallest greater than or equal', @begin\_time);

/\*possible relational operators:

<relational\_operator> ::=

{ largest less than

| largest less than or equal

| smallest greater than

| smallest greater than or equal

}\*/

SELECT @to\_lsn = sys.fn\_cdc\_map\_time\_to\_lsn('largest less than or equal', @end\_time);

select \* from cdc.fn\_cdc\_get\_all\_changes\_dbo\_Employee(@from\_lsn,@to\_lsn,'all')

**Known Issue:** When you call fn\_cdc\_get\_all\_changes\_.. for your capture instance and pass an LSN range that is not valid, we see a error message as: ”An insufficient number of arguments were supplied for the procedure or function cdc.fn\_cdc\_get\_all\_changes\_ ..”.

The issue here is *by design* and is due to the fact that your lower bound is outside the validity interval for the capture instance. The validity interval of the capture instance is indicated by the start\_lsn value in cdc.change\_tables. The ‘from LSN’ that you calculate using the datetime is lower than the start\_lsn. You need to make sure the LSN's are within the validity interval.

The error message is misleading, but it is not possible to raise a meaningful error message as we cannot raise errors from a TVF

This issue has been escalated to **Microsoft Connect** by somebody & the feedback from MS can be found at the URL: [https://connect.microsoft.com/SQLServer/feedback/ViewFeedback.aspx?FeedbackID=333740](https://connect.microsoft.com/SQLServer/feedback/ViewFeedback.aspx?FeedbackID=333740%20)

The workaround suggested by Microsoft is to make sure that the @from\_lsn is greater than or equal to the value specified in the column *start\_lsn* in *cdc.change\_tables* table.

-0-