Introduction to Database Change Data Capture

Table of Contents

Copyright

September 2019

Last updated with new content: Release 12.1

Preface

Purpose

The purpose of this book is to provide an introduction to the Change Data Capture capabilities of the OpenEdge® RDBMS. The book provides an overview of the concepts, terms and definitions, descriptions of capabilities, and pointers to other, more detailed documentation.

Audience

This book is intended for anyone interested in a high-level understanding of Change Data Capture.

Organization

This book is organized into the following chapters:

Change Data Capture on page 9

Presents an overview of Change Data Capture.

Plan for Change Data Capture on page 23

Presents an overview of the planning required for Change Data Capture.

Extract Transform Load (ETL) on page 27

Presents an overview of the Extract Transform and Load (ETL) process.

CDC Schema on page 31

Describes the schema of the CDC tables

CDC Terms on page 39

Defines common CDC terms.

Documentation Map on page 41

Describes where in the documentation set to find additional information.

Documentation conventions

See Documentation Conventions for an explanation of the terminology, format, and typographical conventions used throughout the OpenEdge content library.

1

Change Data Capture

For details, see the following topics:

Change Data Capture Overview

Change Data Capture (CDC) is an industry term that describes the process of duplicating subsets of OLTP data in an external data source with a relatively up to date version of relational data. OpenEdge CDC is an OpenEdge RDBMS feature that identifies and captures data that has changed in tables of a source database, as a result of create, update, and delete (CUD) operations. The OpenEdge implementation of CDC provides a flexible and scalable capture process to facilitate the data extraction, transformation, and eventually the loading of the data to an external data source. This is useful to customers who need to export data to a data warehouse or business intelligence application. The data provided by the capture process exists on the source database and is maintained in relational form.

OpenEdge CDC is flexible because:

- Captured data is maintained in the same database
- Captured data is maintained in relational form
- Captured data is accessible with SQL and ABL
- No changes are required in your application

OpenEdge CDC is scalable because you can define CDC policies such that:

- The amount of data captured is variable by table
- You can capture no data, some data, or the whole record
- You can index the data for easier retrieval

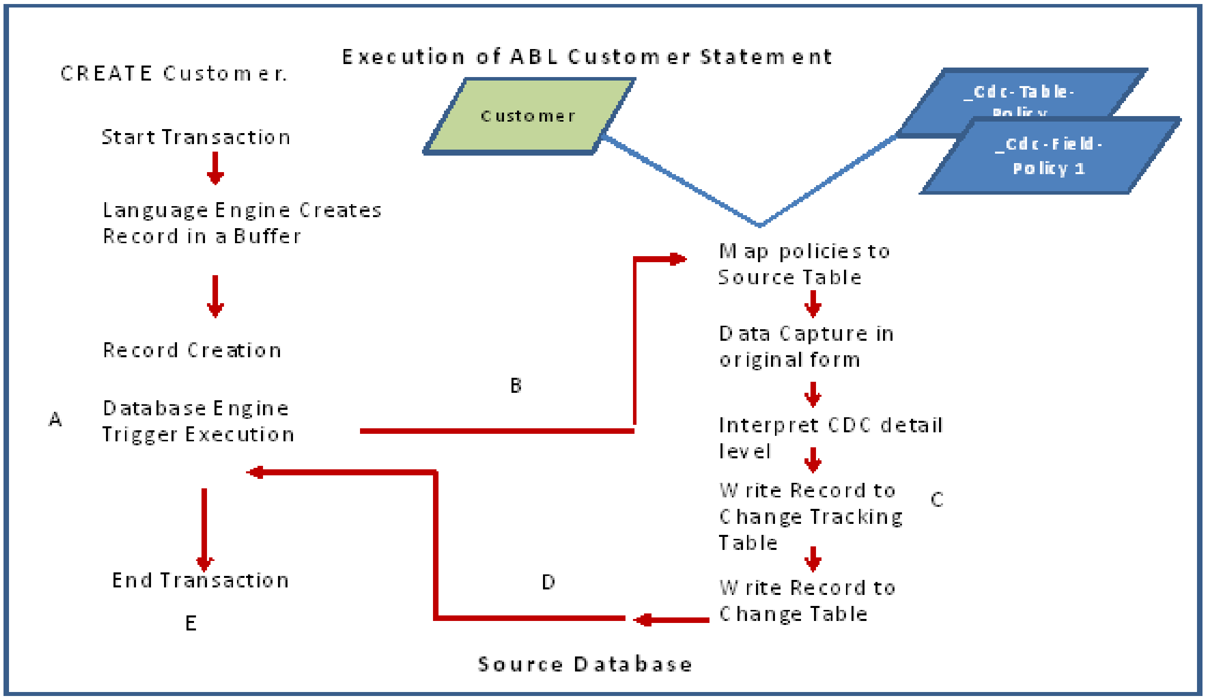
The amount of data captured is controlled through policies defined at the table and field level.

Change Data Capture is a licensed product. It requires either a combination of an RDBMS license (Workgroup or Enterprise) and the corresponding CDC license, or the Advanced Enterprise RDBMS license.

How data is captured

The process of capturing data is driven by CDC triggers which fire for Create, Update, and Delete (CUD) operations. The data captured when the trigger fires is driven by CDC Table and Field policies that must first be created and activated. The triggers execute within a language-initiated (ABL or SQL) transaction, prior to execution of the record activity in the database.

To understand the process, consider the following simple execution flow:


CREATE Customer.
Execution of ABL Customer Statement
Customer
_Cdc-Table-Policy
_Cdc-Field-Policy 1
Start Transaction
Language Engine Creates Record in a Buffer
Record Creation
A
Database Engine Trigger Execution
B
Map policies to Source Table
Data Capture in original form
Interpret CDC detail level
Write Record to Change Tracking Table
C
Write Record to Change Table
D
End Transaction
E
Source Database

When the CREATE CUSTOMER statement executes:

- The original customer record being created is written to the Source data table.
- The CDC triggers in the database engine consult the table and field level policies for the source table being tracked, and data is captured according to the table policy level detail value and field positions selected.
- The trigger first writes a record to the Change Tracking table to indicate transactional scope and sequence.
- The trigger then writes a data capture change record to the Change table based on Table and Field policies.
- Upon transaction commit, all records are committed to the database.

To further understand, the flow, the following terms must be understood:

- **Change Table Policy** — A table used to select a Source table for CDC activity. Table policies are created as a first step in designing what data will be captured from a Source table.
- **Change Field Policy** — A table used to select fields from an identified Source table for CDC activity. A policy represents only one field. A Change Table policy can have several related table policies. The maximum number of field policies per Change Table policy is the number of fields in the Source table.
- **Change Tracking Table** — A table used by the CDC process to insure transactional sequence. This table is created when CDC is first enabled, before any policies are defined. The table represents transactional information for all tables that participate in the CDC process.
- **Change Table** — A user defined data capture table. The construction of this table begins when a CDC policy on the table level is created.

After the execution of one create statement on a table with an active CDC Table policy, the Change Tracking Table is updated, and depending on the level of the Table policy, and the presence of Field policies, the CDC trigger can also write data to the Change Table.

The data captured by the triggers reflects the source records after they are modified. The action of capturing the data is stored in the system wide Change Tracking Table. The captured data is then stored in a source specific Change Table. The Change table is a relational table with data types known to the ABL and SQL. This form of data allows for easy extraction and transformation of the data into an external data store (Data Warehouse). The process of extraction/transformation/loading (ETL) is a user application program.

Tables identified as using CDC need to have their Change Tables monitored for file system growth and be periodically purged. The purge activity can be part of the extraction activity or conducted as a separate activity.

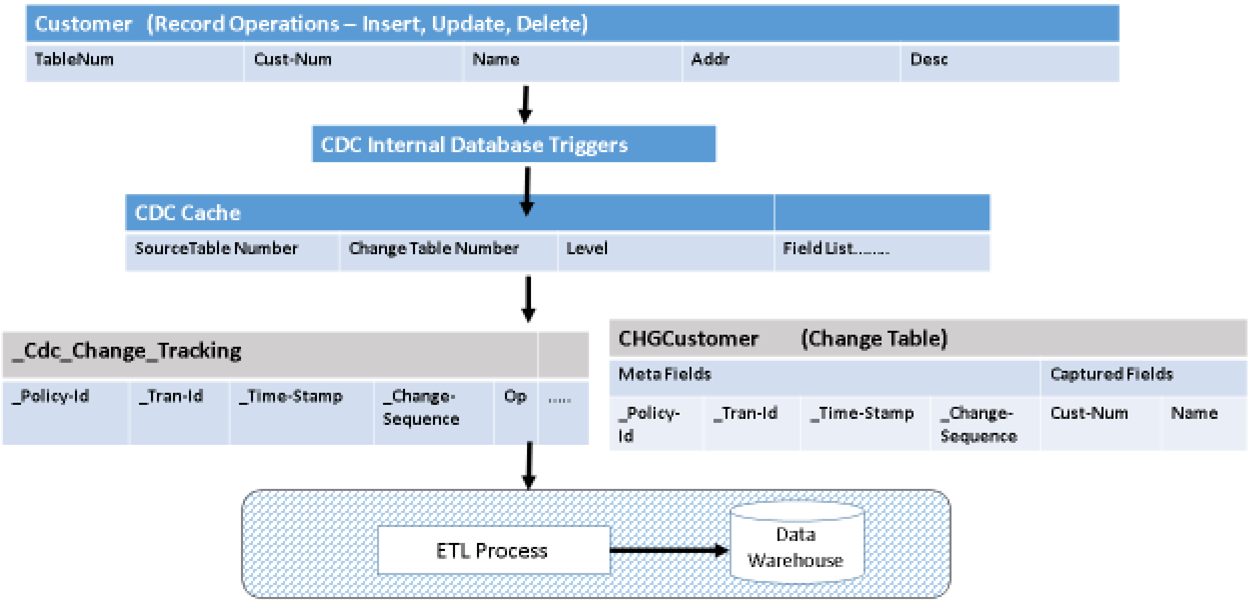
Caution: Managing the growth of the Change Tracking Table and the Change Tables is a very serious issue. The growth of these files cannot be left unmanaged. If either table consumes all the space available to it, the database will shut down until space is added. If the database shuts down, current transactions will be backed out (including the originating transaction).

There may be times when changes for identified CDC tables no longer need to be captured. You can stop capturing change information by deactivating capture activity. This action sets the policy status to inactive. The policies and change data remain in place.

Data Flow

When CDC is enabled, the underlying support for enabling CDC on the table level configured. See Plan for Change Data Capture on page 23 for details on adding CDC to your database. After enabling CDC, a user, referred to as the CDC policy designer, will enable and create table and field level policies to manage the capture of data from the source tables of interest. Once the policy designer has chosen the fields of interest through the creation of CDC Source Table policies, and activated those policies, population of a source table's Change Table begins. When insert, update, or delete record activity occurs, the database engine executes CDC triggers on target source tables. The flow is depicted below.

Figure 1: CDC process flow



The data captured by the CDC triggers reflects the source records after they have been modified. The action of capturing the data is stored in the system wide Change Tracking Table. The captured data is then stored in a source specific Change Table.

The Change Table is a relational table with data types known to the ABL and SQL. This form of data allows for easy extraction and transformation of the data into an external data store (Data Warehouse in the diagram above). The process of extraction/transformation/loading (ETL) is a user application program.

Tables identified as using CDC will need to have their Change Tables monitored for file system growth and be periodically purged. The purge activity can be part of the extraction activity or conducted as a separate activity.

There may be times when changes identified in CDC tables no longer need to be captured. At this time the you can deactivate capture activity in the policy through OpenEdge Management or programmatically. This action sets the policy status to inactive. The policies and change data remain in the database, even when the policy is inactive. See Policy life cycle on page 16 for details on policy status.

CDC Policies

CDC policies guide the production of change data capture records. CDC policy records are managed by the CDC Policy Management Tool in OpenEdge Management or OpenEdge Explorer. Through policies the tool allows the policy designer to manage activities such as schema changes and temporary deactivation of change capture activity for database maintenance. The topics that follow provide more details about CDC policies.

CDC Policy considerations

When you are defining a policy for data capture, the level you set on your Table Policy and the fields you subsequently select to capture with Field Policies, can impact the data captured.

Consider the following when setting the level of your Table Policy. By specifying the Level in the Table Policy, the policy allows flexible capture:

- If you only need to capture the fact the data changed, the Level field can be set to only capture the Change Tracking record.
- If you need to know which fields have changed, but not their values, the change fieldmap in the Change Tracking record can be consulted to see which fields changed. This type of tracking implies that no field policies are created.
- If you only need to track an update operation by recording the current value, the Level field in the Policy table can be set to do this. As a result, the Operation field for an operation type of update indicates “After Update” values are recorded for a change in the Change Tracking and Change Tables.
- If you need to track both the “before update” and “after update” values, the Level field in the Policy table can be set to do this. The Change Tracking Table adds one record to record the operation and the Change Table adds two records, one record for the “before update” field values, and the other for the “after update” field values.
- If a field is deleted from the source table and the field has an active policy, the corresponding field in the Change Table is not deleted. After the deletion of the source fields, the corresponding field in the Change Table stores a null value in the field.
- If all Source fields that have active CDC policies are deleted, there is nothing to be captured, and neither the Tracking nor Change record is recorded.

Consider the following when designing field policies:

- Each selected field generates a field policy.
- The selected fields can be of any basic data type including raw, and array, however LOB columns are not supported.
- The field order (field position) in the Change Table does not have to match the position in the Source Data Table.
- The field order in the Change Table is controlled by the pre-defined meta-data columns as leading fields, followed by identifying fields in their assigned component order, followed by non-identifying fields in policy defined order.
- Identifying fields cannot be of the data types: lob, raw, arrays.
- Selected fields in the Source Data table do not have to be contiguous.
- The fields names in the Source Data table are the source of the field names for the Change Table fields.

CDC Policy Levels

When defining a CDC policy, the level you specify determines amount of change tracking data collected. The following table gives a description of the data captured at level.

Table 1: Description of CDC levels

Level	Description
Minimal (0)	This level indicates a change occurred. No record values will be recorded.
Minimal with Fieldmap (1)	Similar to the Minimal (0) level, this level indicates a change occurred, but also includes a field map value indicating which fields changed. No record values will be recorded.
Medium (2)	This level records the current (after) value of all CUD operations.
Maximum (3)	This level records both the previous (before) and current (after) values of all CUD operations.

The following table describes in more detail the resulting data recorded for the different levels.

Level	Field Policies	Change Tracking Table Record	Change Table Record	Change Fieldmap	Change Level Allowed
0	no	yes	no	null	no
1	yes	yes	no	Reflects changed fields in update only	yes
2	yes	yes	yes	Reflects changed fields in update only	yes
3	yes	yes	• 1 record for before update • 1 record for after update	Reflects changed fields in update only.	yes

Policy life cycle

CDC policies guide the production of change data capture records. In the schema section, the Cdc-Table-Policy and Cdc-Field-Policy tables were defined. The policy designer manages these policies. The OpenEdge Management policy tool allows the policy designer to manage activities such as schema changes and temporary deactivation of change capture activity for database maintenance. The following topics detail the activity surrounding the creation, deactivation, reactivation and disablement of policies by the tool.

Creating policies

Creating a CDC Table and/or its Field policies is when the policy designer establishes user defined indexes and the defines the capture columns of interest in the Change Table. This is also the time the designer indicates the level of capture desired. Creating a policy helps with the transition from one table definition to another due to schema changes. Once the policy set has been established, a certain number of schema change operations are allowed. Other actions allowed for a policy set is to be deactivated, reactivated, or disabled.

Consider how creating a policy on a new table differs from an existing table in terms of captured record operations. The type of record operation captured could affect the initial value of a field in the external data source. Creation of a CDC policy for a new table is different only in the captured field values to start the process are different. In the case of the new table, generating change records could be captured as insert operations in a Change record. CDC capture on an existing table might be captured as update operations in a Change record. The update operations could be of two types: old and new column values in separate Change records. These types of Change records can be used to initialize the external data source during ETL and may yield different results.

Policy Instances

Policies are managed by the value of the _Policy-Instance field in the Table Policy. The table below explains the instance values.

_Policy-Instance Field	Values and their meaning
0	0 means this is a **current** instance of the policy and a Change table may be in existence for this policy depending on policy level. This is the policy that is loaded to the CDC cache and used during data capture by the database triggers.
1	1 means this is a **previous** instance of the policy. If the policy level of the policy required a Change Table, it remains in existence even if the current policy is not adding to it. A previous policy can represent historical data capture.
2	2 means this is a **pending** instance of the policy. The pending policy occurs when a user has not finished editing a policy and will continue at some other point in time.

Only one current instance is in use by CDC triggers for purposes of change capture. Many previous”instances can be in use by the ETL application(s). The ABL language and the Policy Tool will restrict that only one instance of the policy will be a current instance. The previous policies will exist to support data extraction. Purging of previous policy change records is at the discretion of the policy designer. The current policy is used immediately to generate change records through CDC triggers as current CUD operations occur.

Deactivating and reactivating policies

There may be times that particular CDC enabled tables need capture activity suspended temporarily, such as during a time-consuming database maintenance activity. The activity can be temporarily suspended by deactivating a table policy through the policy management tool. Deactivation and reactivation activities apply to policies in active or inactive states (_Policy-State).

Deleting a Policy Set(s)

The deletion of a policy means the policy designer has decided the policy or policies designating that a table is enabled for CDC are no longer needed. This means the object state or instance for each Table policy related to the Source table does not matter. The action will occur on an active or inactive policy set. The action of deleting a policy has the effect of removing the related Change Table and Policy Tables. In addition, the CDC cache entries for these policies are invalidated.

Policies and schema changes

If a source Table is altered by a schema change, a new Table policy may or may not need to be added depending on the need of the policy designer and the intended external data source. When a new field is added to the Source Table, the current policy does not know about the new field, so the policy designer must create a new Table and Field policy set to include this field, if capture is desired. Until the new policy is created, it is as if the field does not exist for the database triggers. When the new field is added and a new Table policy is created, the policy set will now include the new field policy.

A schema change involving a field deletion will cause the current policy to no longer record changes to this column. A data capture for the deleted field will record no value in the change record.

Below are common Source schema changes and how they are supported.

Source Table Schema Changes Supported	Action Required
Change to the Table Policy with an instance of “current” - Level field can be altered to specify either more or less detail in the captured data.	Commit change and it becomes immediately in effect.
Add a field to an existing Source Table. When the new field is added to the Source Table a Change Tracking record with an operation code “add field” is created. In addition, the Change Table’s field map will reflect the newly added fields.	Commit schema change. In order to capture the new field, add a new Table/Field policy set that includes the new field. Upon creation, the new policy becomes the current instance and the old policy becomes the previous instance. As soon as this happens, the database triggers will begin to capture data against the new policy and use the existing Change Table.
Delete a field to an existing Source Table. When the new field is added to the Source Table a Change Tracking record with an operation code “delete field” is created. In addition, the Change Table’s field map will reflect the newly removed fields.	Commit change schema change. From the point the field is removed from the Source Table, the deleted field in the change record will no longer contain values. If a source field participates in an identifying field in the Change Table, it cannot be deleted. The policy(s) containing this field must be deleted (as well as the Change Table itself) before this field can be dropped.
Table Rename of Source Table.	A change of a table name does not require a change to the Table policies because the Source Table recid is stored in the policy, rather than the table name.

Source Table Schema Changes Supported	Action Required
Field Rename of field in Source Table that is being tracked in a field policy.	Field rename is allowed, the field policy does not store field names. The Change Table does use the field name to name the Change fields, so field renames must cascade through the Change Table schema. This action is restricted to an offline operation because it requires a Database Schema lock.
Source table is dropped by Language Drop syntax.	This operation requires that the policies are already deleted. Act of dropping table does not remove records in the Change Tracking Table.
Source partition is dropped	No effect on data capture activity. The capture is based on the Source table and not a specific partition so capture proceeds as usual.
Source partition is renamed.	Both the languages and a utility can rename a partition. The act of renaming a partition does not change the partitionId, so this has no effect on CDC capture activities.

The following table describes supported changes to Table and Field policies.

Table 2: Supported changes to Table and Field policies

Change	Action Required
Change to the Table Policy with an instance of “current” - Level field can be altered to specify either more or less detail in the captured data.	Commit change and it becomes immediate.
If a Table Policy is marked as in the “Pending” instance and the _Object-Id field is set to zero, the table policy can be altered.	Change the table policy. Add/delete field policies. This may occur when a policy was being constructed and was not “finished” and made “current”. This policy can be revisited until it is made to be the current policy.
A field policy may be added, deleted, or changed so long as the parent Table Policy is in the “Pending” instance and its _Object-Id field is set to zero. An object Id of zero means the Change Table associated with the policies have not been created yet.	Change field policy.
Deleting Table/Field Policy Sets. Note: The intention for this operation is that it is a cascading delete operation. All table and field policies and the Change Table will be deleted at once.	Can occur on a Table policy regardless of state or instance. Does not delete related records in the Change Tracking Table.

The following table describes changes to Table and Field policies that are NOT supported.

Table 3: Changes Not Supported for Table and Field Policies

Change	Action Required
If a Change Table was created without an Identifying Field Index, a new Identifying Index cannot be added to the Change Table.	Existing Policy Sets cannot be altered to add a new index. A new policy is required for this table and the old one must be retired.
If a Change Table was created with an Identifying Field Index, the Identifying Field Index cannot be deleted from the Change Table.	Existing Policy Sets cannot be altered to delete an index. A new policy is required for this table and the old one must be retired. An alternative to this situation would be to deactivate the index.
Identifying fields (which are components of the Identify Field Index) cannot be altered or added to an existing Table and Field policy set.	A new policy is required for this table and the old one must be retired.

CDC Tables

CDC tracks and records the data and changes in tables within your database.

- **Schema tables**

  There are two metadata tables: _Cdc-Table-Policy and _Cdc-Field-Policy. These tables are created in the source database when you enable Change Data Capture for the database. The metadata tables contain information about CDC policies for tables and their associated fields. You do not have to allocate space for the metadata tables; these tables reside in the schema storage area of the database. You should not manage these tables, they are managed by OpenEdge RDBMS.

  The relationship between the CDC Table Policy Table and the Field Policy Table is a 1- to-many relationship. The hierarchy consists of two levels: Table policy (_Cdc-Table-Policy) and Field policy (_Cdc-Field-Policy). The table policy is a parent and the field policies are its children.

  The CDC Table policy describes the policy attributes such as filename of the source table reporting to this policy, area the table resides in, active status of policy, and coverage level of the policy. This policy can point to a number of CDC Field policy records that identify the fields to be captured in the Change Table for this table. Once the new policy is defined, created, and committed, the policy becomes active. Each time a CUD record operation is executed, a record is written to the Change Tracking Table and the Change Table assigned to the policy.

- **Data tables**

  There are CDC data tables: the _Cdc-Change-Tracking table, and an individual CDC Change Table for each source table for which you want to capture change data. Database administrators are responsible for allocating space for all CDC data tables, and specifying the area when enabling CDC.

  A CDC Change Table is created in the source database when you define and activate a CDC policy for a source table. Note that a CDC Change Table is created only when the policy level is higher than minimal. CDC policies with the minimal level do not have a Change Table; their data is written only to the _Cdc-Change-Tracking table.

  The _Cdc-Change-Tracking table maintains a record of all change capture activities for *all* tables with CDC policies. This table is created in the source database when you enable Change Data Capture for the database.

See CDC Schema on page 31 for details of the schema for these tables.

Naming Change Tables

Change Tables can be given a name when the policy is designed, or a generated name is suggested at policy creation in the OpenEdge Management. Tables created through the ABL can be named by the policy designer. Change Table names can be a maximum of 32 characters and must be unique with their schema domain. The Change Table name must follow current table naming conventions.

If you leave the name blank when creating a policy, the table name is set using the default naming. In most cases the default naming is a good choice since it clearly identifies the table as a CDC table and it includes the name of the source table. The default naming appends "CDC_" to the name of the source table. For example, when the source table is Customer, the default Change Table name is CDC_Customer.

Change Table ownership and location

Tables created by the ABL are owned by the PUB schema and tables created by SQL can be owned by the PUB schema or another owner. The ABL can only drop tables in the PUB schema and SQL can only drop tables in the PUB and non-PUB schemas. Change Tables present an issue of ownership when they are created. A policy designer should consider the language ant ETL application is written in to best determine Change Table ownership upon creation. Access to the Change Table is controlled by the owning schema, this is particularly important for the ability to drop this table.

The policy designer needs to indicate Change Table ownership in the _Cdc-Table-Policy Table in the field _Ownership. The _Ownership field allows the designer to indicate ownership by specifying that ownership of the Change Table will be the same as the Source Table or that ownership will be PUB.

In addition, the Change Tables reside in a Type II area specified in the Table Policy for the table. This means the Source Table could reside in a Type I area and its Change Table will reside in a Type II area. The area location of the change files is modifiable after creation by the use of Table move. The name of the Change Table is not modifiable. If a new name is desired, a new policy will need to be added.

Life cycle of a Change Table

Once a Change Table is created for capture from a Source Table, it is possible for it to represent many policies over the course of its lifetime.

Many previous policies can come from a range of schema changes allowed:

- Add new fields to the Source Table
- Add policies for new or existing fields to the Change Table
- Delete Source Table fields
- Delete Field policies

Changing identifying fields (and by association their index) is not a permitted schema change. If this activity is required, CDC activity must be disabled for the table of interest and all change data removed. A new policy must be created to add the desired identifying fields and index.

Operations allowed on CDC tables

To maintain the integrity of CDC tables some operations on these tables are not allowed by users. The following table provides details.

Table 4: Operations allowed on Change Tables and Change Tracking Tables

Operation	Change Table	Change Tracking Table
Read	Allowed	Allowed

Operation	Change Table	Change Tracking Table
Delete	Allowed	Allowed
Create	NOT allowed	NOT allowed
Update	NOT allowed	Allowed on _User-Misc field NOT allowed on any other field

You can read and delete records, provided they have the proper authorization. For example, it is a common process for an ETL application to read records to copy them to an external data source, or a DBA may delete change table and change tracking table data when it is no longer needed. However, you are not allowed to create records in these tables, and with one exception they are not allowed to update records.

The process of capturing the data is handled by an internal OpenEdge RDBMS trigger. The creation of the records is *exclusively* in the domain of the internal CDC database triggers.

Users are allowed to update just one field in the Change Tracking Table record. Only the _User-Misc field can be modified. This field is available to allow runtime customization by applications that consume the CDC data. For example adding data to this field to indicate that a record has been read or that it can be deleted.

Retrieve data

After recording data changes, the data needs to be accessible in an efficient and orderly manner.

The ordered access comes through the use of the indexes that are provided on the change tracking table and on the CDC change tables.

There is a relationship between the records in the change tracking table and the change tables,each change tracking table record has zero, one, or two associated change table records depending on the Level of the CDC policy.

The connection between these two tables is important. The type of change activity (operation) is in the Change Tracking Table, and the data resulting from the operation is in the change table. Each time the record is modified, there is a record added to the Change Tracking Table based on the operation. Operations that are tracked, include: create, update, and delete. See Change Tracking Table on page 35 for a complete list of operations.

The operations are ordered based on change sequence order field. The change sequence field has an increasing value as records are added to the table.

When extracting the change data, it often needs to read the data in the order it occurred, either by the sequence of entry, or by time of entry. Indexes ease this process.

- The _Change-Sequence-Id index establishes a (foreign key) relationship to the Change Tracking Table.
- The _Time-Sequence index allows access to a change table through the combination of change sequence and a timestamp fields.
- Optionally, the _Identifying-Fields index.

  A CDC policy designer selects the fields and component order for this index when defining a policy. A final component is always added after the identifying fields , the _Change-Sequence field to preserve the order.

Split Records

As data is written to the Change Tables, it is possible that the Change Table record can exceed the maximum OpenEdge database record size. When this occurs, two Change Table records are written for one change. The second record is known as a split record, or continuation record, since the change is split across two record.. As a part of the CDC implementation, whenever the data exceeds what can be stored in one Change Table record, the database automatically handles creating the two Change Table records to hold the data. The database stores information in the meta-data columns that is needed to return one record, in place of these two records, when the change data record is read. It again fits back into one record, since with the change data is read the meta-data is not returned.

A split record occurs is when data is copied from fields in the source table to the Change Table, and the total size of CDC user data for all captured fields combined is very near to the 32,000 byte limit. Although the meta-data columns don't consume very much space relative to the record size, the addition of these columns can push the total record size over the size limit.

SQL and ABL handle reading split records differently.

- **SQL**

  The SQL engine automatically handles returning one record when reading CDC Change Table record data that has been split into two records. In other words, when coding SQL you don't need to be aware whether or not there is a continuation record.

- **ABL**

  ABL does not automatically handle split records. The ABL programmer must rejoin the split records in code. Helper classes are provided in the `OpenEdge.DataAdmin.Util` package to aid this process. See *OpenEdge Programming Interfaces* for details.

2

Plan for Change Data Capture

Before you can implement Change Data Capture, you need to evaluate your database and tracking needs. You will need to identify source tables and fields for CDC, the CDC level for each policy, and how many storage areas you will need for CDC change tables and indexes.

As you plan your implementation of CDC, consider the following questions:

- For which source tables and fields do you need to capture change data?
- Which fields of the source tables should be used to create indexes?

  In CDC, these fields are referred to as identifying fields. Identifying fields speed up the process of extracting data from the CDC change tables and help to quickly locate related data in the data warehouse for updates.
- Which CDC levels should your policies have?
- How many storage areas will you need for your CDC data tables and their associated indexes?

For details, see the following topics:

- Allocate space
- Enable CDC
- Create CDC Policies
- CDC User Identity

Allocate space

Allocating space for CDC data tables and indexes ensures that CDC data tables and indexes have adequate space for current usage and future growth. Type II storage areas are required for all CDC data tables and indexes.

Change tracking table

The _Cdc-Change-Tracking table is the repository of all change capture data in a database. Once Change Data Capture is running, _Cdc-Change-Tracking can grow very quickly and become very large. It is important that you allocate enough space for this table and its indexes.

Change tables and indexes

A change table is created for every source table you define a CDC policy for with a policy level is higher than minimal (0).

Because each source table differs from other source tables in record sizes, number and data types of fields, and growth patterns, you need to make decisions about storage areas for your CDC change tables and indexes.

Consider the following guidelines to create storage areas for individual CDC change tables and their indexes:

- For each CDC change table that is expected to be frequently updated, create one storage area for the table.
- For CDC change tables that are expected to be less frequently updated and that share similar characteristics (same data type, similar records per blocks or blocks per cluster), create one storage area for each category of tables.
- For the indexes of each CDC change table that is expected to be frequently updated, create one storage area for the indexes.
- For the indexes of CDC change tables that are expected to be less frequently updated and that share similar characteristics, create one storage area for each category of indexes.
- For storage areas of CDC change tables, use:
    - 512 blocks per cluster for high growth
    - 64 blocks per cluster for moderate growth
    - 8 blocks per cluster for low growth
- For storage areas containing only indexes associated with CDC change tables, use:
    - 1 record per block
    - 64 blocks per cluster

Enable CDC

Before you can use Change Data Capture to track your changes, you must enable your database. There are two ways you can enable your database:

- Using PROUTIL

You can use the PROUTIL ENABLECDC qualifier. The syntax is:

```
proutil db-name -C enablecdc
     area table-areaname indexarea index-areaname
     [ deactivateidx ]
```

See *Manage the OpenEdge Database* for details.

- You can use the Database Administration Console of OpenEdge Management or OpenEdge Explorer. See *Manage Database Change Data Capture with OpenEdge Management* or the online help for details.

Create CDC Policies

Capturing data with CDC requires you to create policies that define the tables and fields that are tracked, and the amount of change recorded. There are two ways to create policies:

- **Create Policies with the Database Administration Console of OpenEdge Management or OpenEdge Explorer**

  The Database Administration Console provides you with a graphical user interface to define policies. For more information see, *Manage Database Change Data Capture with OpenEdge Management* or the online help for details.

- **ABL API**

  The ABL API for CDC is a set of ABL user-defined classes and interfaces that enables you to access and update the schema of a database without having to know the details of the ABL API. To make your task easier, OpenEdge enables you to generate a CDC policy program as a template that you can modify to create CDC policies. To generate this program, you must have at least one existing CDC policy for a source table. For more information see, *OpenEdge Programming Interfaces* for details.

CDC User Identity

OpenEdge provides the option of identifying the user making a change within a Change Tracking Table record.

The user's identity is stored in the `_User-Name` field of the `_Cdc-Change-Tracking` table. There is no schema change.

The user's identity is set to one of the following options:

- **None** — Does not record any user identity.
- **Database** — Records the user's identity based on the database connection used by the user.
- **Application** — Records the user's identity based on the application session used by the user.

Configure the recording of the user's identity with OpenEdge Management or with a program authorized and written to manipulate the `_Db-Options` table. For details configuring the user's identity with OpenEdge Management, see *Manage Database Change Data Capture with OpenEdge Management* .

Configuration information

When configuring the CDC User Identity, the following restrictions apply:

- If at least one Security Administrator exists, then only a Security Administrator can change the user identity recorded in the Auditing and CDC Change Tracking tables.
- If no Security Administrator exists, only users with appropriate access to the `_Db-Options` table can change the user identity recorded in Auditing or CDC Change Tracking tables.
- The Security Administrator, or user with `_Db-Options` access, can request any of the following options for how the user identity is recorded in the Change Tracking Table:
  - None (this is the default behavior).
  - The user identity with no domain information (shortened user identity), for example `user1`.
  - The user identity with the domain information (fully qualified user identity), for example `user1@companydomain`.

  Note: This is provided implicitly and cannot be overridden if the database is enabled for multi-tenancy.

  Where optional, the reporting of a shortened or fully qualified user identity is database wide. However, when both Auditing and CDC are enabled, you cannot configure the user identity as separate options for each feature.

  For Auditing and CDC, the interactions are as follows:

  - Auditing does not need to be enabled to specify the qualification of the CDC user identity reported.
  - CDC does not need to be enabled to specify the qualification of the Auditing user identity reported.
  - However, either Auditing or CDC must already be enabled to specify this option at all.

Changing Identity

Although the changes to the CDC identity can be made online, the change only takes effect at the next occurrence of a SETUSERID() / SECURITY-POLICY:SET-CLIENT() / SET-DB-CLIENT() implicit or explicit action. This behavior is consistent with auditing.

If application session identity is requested, and the application has not set application session identity for the client connection, then the database connection identity is recorded.

Related Information

For more information user identity, see:

- The Application Security section of *OpenEdge Programming Interfaces*.
- The Audit Security section of *Learn about Security and Auditing*.
- The Configuring and Implementing Authentication in OpenEdge section of *Learn about Identity Management*.

3

Extract Transform Load (ETL)

ETL is the acronym for the "Extract, Transform, and Load" process.

For details, see the following topics:

- ETL defined
- ETL and Schema changes
- Language support for ETL

ETL defined

ETL is a process for executing the three steps in its acronym.

ETL "tools" combine these three functions :

- Data is *extracted* from one database and *loaded* into another database
- A *transformation* may be made to the data, based on the needs of the external application, before it is loaded into the external data repository

ETL "tools" are free-form, and can divide and combine these three concepts or functions into as few or as many steps or programs as required to complete the process of moving data to an external data warehouse.

Companies can write their own ETL tools, or choose from one of the tools that are available in the market place.

The extraction, transformation, and loading of CDC data can be done using OpenEdge ABL, OpenEdge SQL, or a third party tool. With CDC you can increase ETL efficiencies through the accurate identification, tracking and storing of changes. There is also no need for downtime of the OpenEdge RDBMS to extract the data, or changes to the application to save it.

ETL and Schema changes

Schema changes can impact your ETL process. The following table describes the impact of various schema changes.

Table 5: Schema change impacts

Source Table	Change Table (C)	Change Tracking Table (CT)	ETL
Add a field	No record	No record	No action required.
Add a field policy which is a new field in source table	New field	Record written indicating Field Add	Existing ETL program continues to work. You can use CT record to know if this action occurred and decide to alter ETL.
Delete field (offline)	Rename field on Chg Tbl (offline) causes CRC issue on ETL process **	Record written indicating Field Delete	Existing ABL ETL will fail with a CRC issue immediately. If SQL ETL references the deleted field name it will fail. There can be dynamic use of the CT record indicating field delete.
Delete field policy (Create a new policy without a field), source field still exists	Field exists with old name	Record written indicating Field Delete	ETL can use bitmap to indicate which fields can be extracted.
Re-add field policy with change field already in Change Table	Field exists	Record written indicating Field Add	Existing ETL program continues to work. You can use CT record to know if this action occurred and decide to alter ETL.
Add field policy with existing field name in source table	New field in the change table	Record written indicating Field Add	Existing ETL program continues to work. You can use CT record to know if this action occurred and decide to alter ETL.

Source Table	Change Table (C)	Change Tracking Table (CT)	ETL
Re-add field and field policy with same name as was previously dropped	New field with the same name added(online)	Record written indicating Field Add	Use CT record to know if action occurred
Rename field on source same as Change field that was deleted with ABL special naming convention			No action
Rename field on source that already has a change field	Field renamed		Existing ETL will be effected by a CRC change caused by rename.

Language support for ETL

One of the strengths of OpenEdge CDC is that the Change Tracking Table and the data Change Tables are standard tables in the database. You can write ABL or SQL to handle the CDC tables, just like all the other tables in your database. For information on language support and brief examples, see:

- **For SQL** — See *Develop SQL for OpenEdge*
- **For ABL** — See *OpenEdge Programming Interfaces*

4

CDC Schema

The topics that follow describe the schema CDC tables.

For details, see the following topics:

- CDC Policy Table
- CDC Field Policy Table
- Change Tracking Table
- Change Table

CDC Policy Table

The following table describes the fields of the _Cdc-Table-Policy.

Table 6: _Cdc-Table-Policy schema

Field Name	Data Type	Description
_Policy-Id	char	Unique CDC policy id
_Policy-Name	char	Policy name
_Policy-Desc	char	Policy description

Field Name	Data Type	Description
_Policy-State	integer	CDC policy state • 0 — Active • 1 — Inactive
_Policy-Instance	integer	CDC assigned instances • 0 — Current policy (currently capturing data) • 1 — Previous policy • 2 — Pending policy
_Source-File-Recid	dbkey	Dbkey of the source table
_Area-Ianum	integer	Initial area to store the CDC Change Table. Areas must be Type II.
_Area_Index-Ianum	integer	Initial area to store CDC Change Table indexes. Areas must be Type II. If not specified, indexes default to the same area as the Change Table.
_Change-Tablename	char	Filename of the Change Table
_Object-Id	integer	File number of the Change Table
_Identifying-Fields	logical	Does the user want to define identifying fields and create an index for them? (Y/N)
_Level • A level setting of 0 is not modifiable once a policy has been activated. • All levels record entries in the Change Tracking Table. • Levels 1 and above cause creation of the Change Table. • Levels 2 and above record values in the Change Table.	integer	Defines how much detail a user wants to capture in the change record. • 0 – (minimal tracking) No record values are recorded. This level of tracking indicates a change occurred. A Change Tracking Record is generated and the fieldmap of fields is empty. **No Change Table is created for this level**. This level cannot be changed/upgraded to a higher level once this policy is active and committed. • 1 – (minimal tracking)

Field Name	Data Type	Description
		Similar to level 0 minimal tracking, level 1 of tracking only generates a Change Tracking Record, but it includes a fieldmap indicating which fields changed. It also does not include a Change Table record. **Unlike level 0, a Change Table is created for compatibility moving to higher level policies**. • 2 – (Medium tracking) All operations record the current value only. • 3 – (Maximum tracking) Both previous value and current value are recorded (Applies to record update only).
_Change-Table-Owner	char	Default value for the Change Table owner is PUB. If another owner is desired, specify here.
_Encrypt-Policy	logical	Default to encrypt Change Table if Source Table is encrypted. (Y/N)
_Last-Modified	timestamp tz	Time stamp reflecting when the _Policy-Instance changed. Changes occur when: • The policy is first created • The policy becomes current • The policy becomes previous.
_First-User-Field	integer	Beginning of user data fields in Change Table.
_Misc	char	Reserved for future use.

The following table describes the indexes of the _Cdc-Table-Policy.

Table 7: _Cdc-Table-Policy indexes

Index Name	Index Field
Index Name	Index Field
_Policy-Id	1 _Policy-Id

Index Name	Index Field
_Policy-name	1 _Policy-Name
_Policy-Source-Recid	1 _Owner-Name 2 _Source-File-Recid

CDC Field Policy Table

The following table describes the fields of the _Cdc-Field-Policy.

Table 8: _Cdc-Field-Policy fields

Field	Size	Description
_Policy-Id	char(22)	Relationship to parent
_Field-Position	integer	Physical position of field in the source table
_Identifying-Field	integer	A non-zero value ranging from 1 to 15 indicating ordinal position in the index.
_Field-Recid	int64	Source field recid (avoids update of policy on field rename)
_Misc	char	Reserved for future use

The following table describes the indexes of the _Cdc-Field-Policy.

Table 9: _Cdc-Table-Policy indexes

Index Name	Index Field
_Policy-Id	1 _Policy-Id 2 _Field-Position
_Identifying-Field	1 _Policy-Id 2 _Identifying-Field
_Field-Recid	1 _Policy-Id 2 _Field-Recid

Change Tracking Table

The following table describes the fields of the _Cdc-Change-Tracking table.

Table 10: _Cdc-Change-Tracking fields

Field	Size	Description
_Policy-Id	char(22)	Table Policy Id
_Tran-Id	integer	Transaction number
_Time-Stamp	timestamp tz	Timestamp of database trigger execution – UTC with zone
_Change-Sequence	int64	Sequence of change within the Object for a policy Id
_Operation	integer	Value to indicate what type of operation is occurring: • 1 — Create • 2 — Delete • 3 — Before Update • 4 — After Update • 5 — Field Delete (in Source Table) • 6 — Partition Drop • 7 — Partition Truncate • 8 — Partition Deallocate • 9 — MT Partition Deallocate • 10 — MT Partition Drop • 11 — Merge Partition Drop • 12 — Field Add (in Change Table, also specifies re-tracking of a field)
_Source-FieldMap	raw(256)	ABL/SQL methods are provided to access values in this field.
_Source-Rowid	int64	Support for Pro2
_Source-Partition-Id	integer	Support for TP
_Tenant-Id	integer	Support for MT

Field	Size	Description
_Version	integer	Future to support change table record format changes
_User-Misc	char x(8)	User defined field — possible use to mark it when an operation has occurred that moved a record to the external data source.
_Misc	char x(8)	Reserved for future use

The following table describes the indexes of the _Cdc-Change-Tracking table.

Table 11: _Cdc-Change-Tracking indexes

Index Name	Index Field	Attribute
_Sequence-Id	1 _Source-Table-Number (These two fields together point to a single instance of the table) 2 _Change-Sequence	Unique Primary
_Time-Stamp-Seq	1 _Source-Table-Number 2 _Time-Stamp 3 _Change-Sequence	Non Unique
_Part-Rec-Id	1 _Source-Table-Number 2 _Partition-Id 3 _Recid 4 _Change-Sequence	Unique

The CDC Change Tracking Table is a table used by all change record capture activities. The table tracks each CUD (create/update/delete) activity by a change sequence order. This table can be used to drive query activity in the CDC Change Table. The database CDC trigger will write a Change Tracking and Change Table record in most cases when it is executed. Understanding the connection between the two tables is important because data describing the change activity is stored in the Change Tracking Table. When accessing the the data in the Change Table, use the Change Tracking Table to establish record modification sequence and which fields changed in the Source Table.

Change Table

The Change Table schema has two parts: metadata fields to tie the change table to the original record, and column data specified by the policy definition.

The metadata portion of the change table contains the fields described in the table that follows.

Table 12: Change table metadata

Field	Size	Description
_Tran-Id	integer	Transaction number used to create this record.
_Time-Stamp	timestamp tz	UTC timestamp and zone of the database server.
_Change-Sequence	int64	Number indicating an operation's capture order within an object.
_Continuation-Position	integer	Support for larger than 32K record sizes. • 0 — no continuation • Nonzero indicates last successfully processed field.
_ArrayIndex	integer	Used in conjunction with _Continuation-Position to identify an element of an array field. If a Non-zero value, it is the array index of the element.
_Fragment	integer	This value means a fragmented field is either present or not. • 0 —This field is not split into 2 records. • 1 — This field is split into 2 records.

The remainder of the record, the column data, is defined by the source table fields defined in the CDC policy. The general format is as follows:

User Data Columns specific to Source Table. The name of the field is taken from the Source table field that was selected.	Source Field Data type	Value from Source Table at the time of capture

The indexes for the Change Tracking tables are described in the following table.

Table 13: Change Table indexes

Index Name	Index Field	Attribute	Description
_Change-Sequence-Id	1 _Change-Sequence 2 _Operation 3 _Continuation	Unique Primary	A CDC mandated index to establish a (foreign key) relationship to the Change Tracking Table. This index provides easy access to a chain of change records through the _Change-Sequence field.
_Identifying-Fields	1 User Defined Column Through 15 User Defined End Column 2 _Change-Sequence	Non Unique	Optional index created when the policy is defined and fields are picked to become part of the the identifying field index (_Identifying-Fields). This index is completely defined when the policy is designed, including index component order. A final component is added to the identifying field index, the _Change-Sequence field to preserves order.
_Time-Sequence	1 _Time-Stamp 2 _Change-Sequence	Non Unique	A CDC mandated index that allows access to the Change Table through a combination of change sequence and a timestamp. This index promotes time sequencing of changes within a table (object).

5

CDC Terms

The following terms are used in the discussion of Change Data Capture:

- **Change Data Capture (CDC)** – Industry term used to describe the duplication of subsets of OLTP data in an external data source with a relatively up to date version of relational data. Implementations of CDC can involve an extraction/transformation/load (ETL) application that captures individual data changes per table. Instead of dealing with the entire table, the ETL application can selectively choose which fields from an individual table to capture and to populate to an external data source.

- **Change Field Policy** – Describes a table used to select fields from an identified Source Table for CDC activity. A policy represents only one field. A Change Table policy can have several of these related policies reporting to it. The maximum number of field policies per Change Table policy is the number of fields in the Source Table.

- **Change Record Sets** (also referred to as chained Change records) – Describes the splitting of change data (which represents one change) into a chained grouping of records. The chaining of these records is indicated by the use of the continuation position field.

- **Change Table** – Describes a user defined data capture table. The construction of this table begins when a CDC policy on the table level is created.

- **Change Table Policy** – Describes a table used to select a Source table for CDC activity. The policy designer through the OpenEdge Management Policy Tool creates table policies as a first step in designing what data will be captured from a Source table.

- **Change Tracking Table** – Describes a table used by the CDC process to insure transactional sequence. The construction of this table occurs when CDC is enabled. The table represents transactional information for all tables that participate in the Change Capture process. Their participation begins when a source table is enabled for CDC through the OpenEdge Management Policy Tool.

- **Identifying fields** – Describes the fields the CDC policy designer uses to potentially build an index in the external data source table being constructed from the Change table data. The CDC policy designer must select from the source data file, the fields that best identify each particular change.

- **Identifying index** –Describes an index containing identifying field values used in identifying/locating records in the external data store. Once the CDC designer has created the identifying fields, the CDC capture process will create the record and related index using those values from each source data record.
- **Meta data -** Describes the columns in the Change Table Header. Meta data means data which describes data. The initial columns in the Change Table are meta data columns containing information describing data in the Change Table that is necessary for understanding source fields changed and whether record/field splits are present.
- **Source Table** – Describes a table that contains original data to be captured by the CDC process. The table can be of any user defined schema table. Virtual, System, View, and non-user defined schema tables are not eligible to participate in the CDC process. Each source table must be enabled individually through the OpenEdge Management Policy Tool.
- **Table Policy Record Sets** – Describes the grouping of a Table policy record with its related Field policy records by use of a foreign key. This grouping implies that they are considered a unit and need to be managed as a unit.

6

Documentation Map

This topic provides a documentation map that directs you to additional sources of more detailed information regarding Change Data Capture in OpenEdge.

Table 14: CDC documentation map

For additional information on...	See...
Database Administration	The manual, *Manage the OpenEdge Database*. The Database Administration Utilities online help.
CDC management in OpenEdge Management and OpenEdge Explorer	The manual, *Manage Database Change Data Capture with OpenEdge Management*
CDC Policy management	The manuals: • *Manage Database Change Data Capture with OpenEdge Management* • *OpenEdge Database Tools* The OpenEdge Management and OpenEdge Explorer online help, and Data Dictionary and Data Admin online help.

For additional information on...	See...
ABL for CDC	The manual,*OpenEdge Programming Interfaces*
SQL for CDC	The manuals: • *Develop SQL for OpenEdge* • *OpenEdge SQL Reference*