

Change Data Capture

Date: May 2025



Change Data Capture (CDC) involves the identification and capturing of changes made to data within a database. This allows for real-time data integration and synchronization across different systems so the most recent data is always available for analysis, reporting and other business functions. The strengths of OpenEdge CDC, include its flexibility and scalability.

OpenEdge CDC is flexible because:

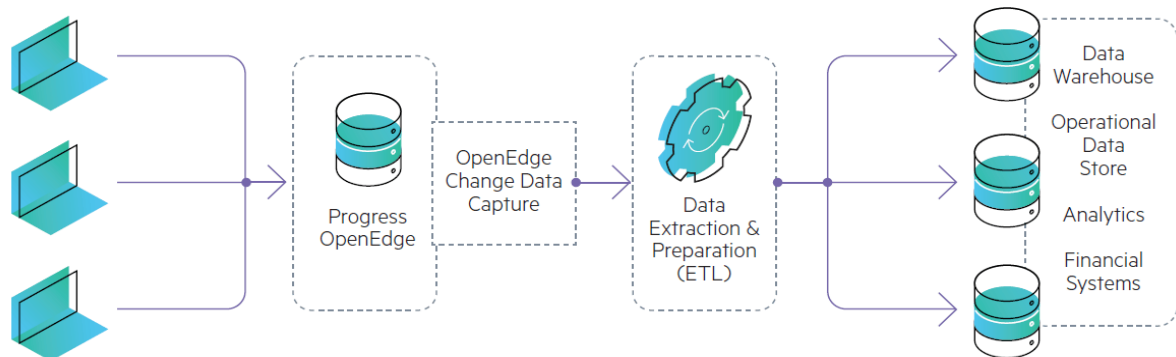
- Captured data is maintained in the same database
- Captured data is maintained in relational form
- Captured data is accessible with SQL and ABL
- No changes are required in your application

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.



The main characters of the CDC include:

Versatility

- Records of changes are stored in the same database.
- Records of changes are stored in relational form.
- Changes records are available by standard SQL, ABL resources.

Good scalability

- The rules allow you to determine the tables and their vertical subsets (columns), the changes of which will be recorded.
- Data in Change Tables can be indexed.
- You can control the number of changes saved.

High utility value

- Tracking only the required changes in the OpenEdge database.
- Saving changes in one place, regardless of where the original data occurs.
- A suitable data source for ETL (*Extract, Transform, Load*) tools, DW and other data storage.
- Single configuration regardless of the way of access (SQL, ABL).
- CDC is completely independent of applications.
- Online management without stopping the database server.

When CDC is activated, two tables are added to the core metaschema: **_Cdc-Table-Policy** and **_Cdc-Field-Policy**. These contain information about CDC rules for tables and their fields. Additionally, a "Change Table Tracking" table (**_Cdc-Change-Tracking**) is added. Once CDC is configured, a separate change table is created for each database table being tracked—this is where the recorded changes are stored.

Table **_Cdc-Change-Tracking** stores records of all activities related to data changes in the original tables, based on the defined CDC rules. Changes are captured using special CDC triggers, and are integrated directly into the database engine. This table also maintains the correct sequence of database transactions. The actual change data is written to the appropriate Change Table via these triggers. An optimization mechanism ensures that the amount of stored data can be limited according to the configured rules.

Before activating the CDC function, you must create an area for new tables and indices. Due to the frequency of changes, this must be space sufficiently dimensioned. An important task of the database manager is to monitor the growth of this data.

Example:

Now let's look at a practical demonstration. Based on the standard sports2000 database, we create a database ready for CDC. Remember that to run the CDC, the database must contain Type II areas.

First, we create a new database, such as myCDC, a copy of the sports2000 database with the command:

prob myCDC sports2000

Then add the areas (areas) in which the CDC data will be stored. First, we create a file **add_CDC.st** :

```
#
d "CDC_Track_Data":20,64;512 .
#
d "CDC_Track_Index":21,1;64 .
#
d "CDC_Customer_Data":25,64;512 .
#
d "CDC_Customer_Index":26,1;64 .
#
d "CDC_Misc_Data":27,64;512 .
#
d "CDC_Misc_Index":28,1;64 .
#
```

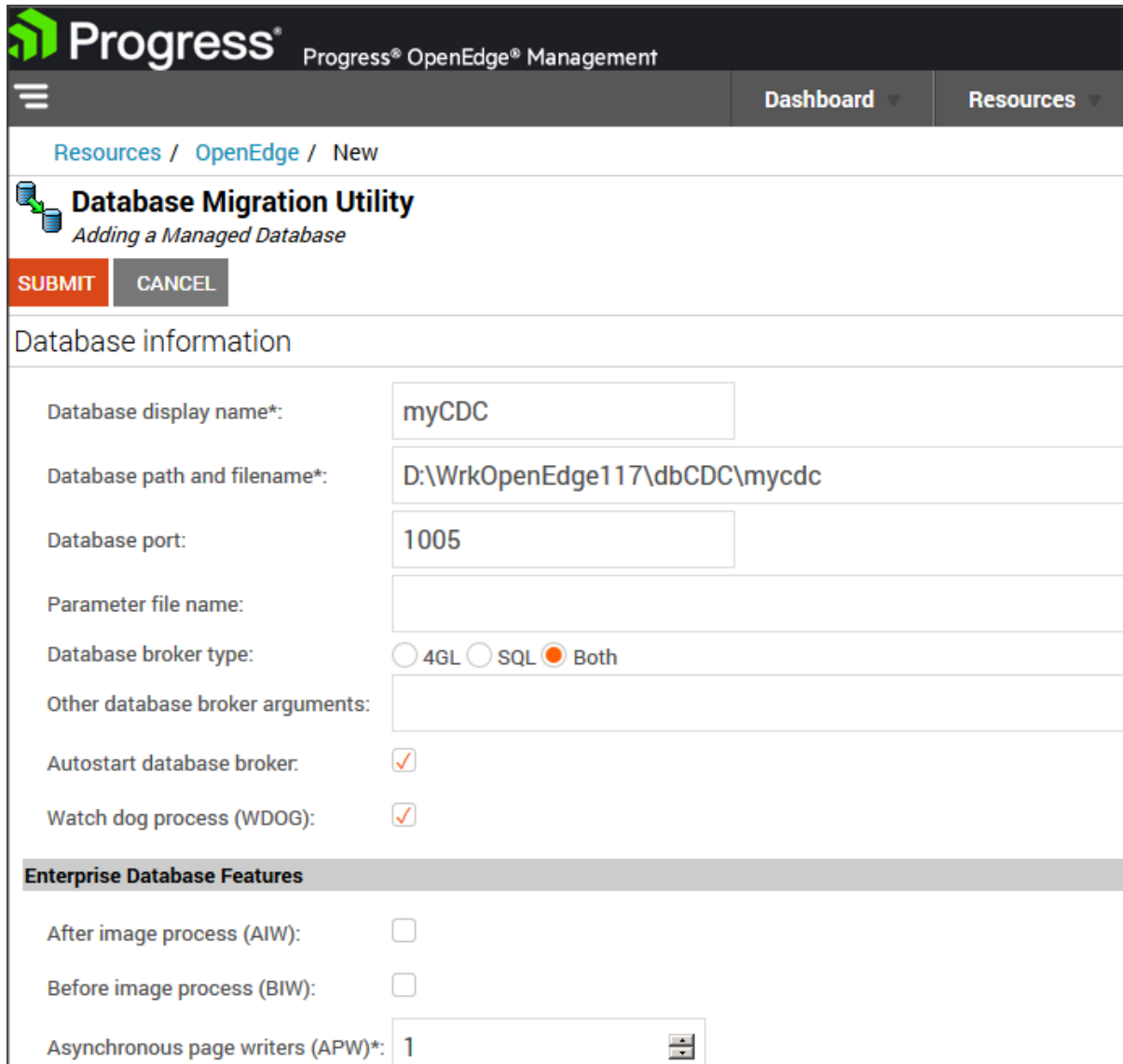
From the command line, **we** create new areas:

prostrct add myCDC add_CDC.st

```
proenv>prostrct add mycdc add_CDC.st
OpenEdge Release 11.7.2 as of Tue Oct 24 19:02:01 EDT 2017

Converting relative path database to absolute path database. <8461>
Formatting extents:
size      area name      path name
512        CDC_Track_Data  D:\WrkOpenEdge117\dbCDC\mycdc_20.d1 00:00:01
64         CDC_Track_Index D:\WrkOpenEdge117\dbCDC\mycdc_21.d1 00:00:00
512        CDC_Customer_Data D:\WrkOpenEdge117\dbCDC\mycdc_25.d1 00:00:00
64         CDC_Customer_Index D:\WrkOpenEdge117\dbCDC\mycdc_26.d1 00:00:00
512        CDC_Misc_Data   D:\WrkOpenEdge117\dbCDC\mycdc_27.d1 00:00:00
64         CDC_Misc_Index   D:\WrkOpenEdge117\dbCDC\mycdc_28.d1 00:00:00
```

For the next steps, we'll use the OE Explorer or OE Management web interface. First, we'll register the newly created database. After logging in, go to **Resources → Databases**. This opens the *Migration Utility* screen. Here, enter the connection parameters for the myCDC database, including the port number (e.g., 1005). Make sure to check the **AutoStart database broker** option.



Progress® OpenEdge® Management

Resources / OpenEdge / New

Database Migration Utility

Adding a Managed Database

SUBMIT **CANCEL**

Database information

Database display name*: myCDC

Database path and filename*: D:\WrkOpenEdge117\dbCDC\mycdc

Database port: 1005

Parameter file name:

Database broker type: ☐ 4GL ☐ SQL ☒ Both

Other database broker arguments:

Autostart database broker: ☒

Watch dog process (WDOG): ☒

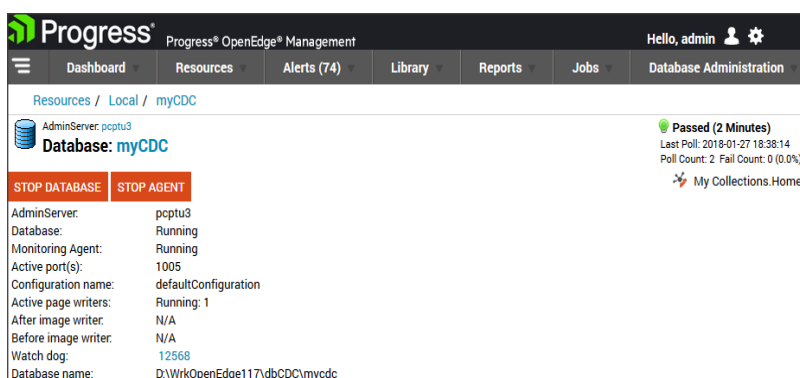
Enterprise Database Features

After image process (AIW): ☐

Before image process (BIW): ☐

Asynchronous page writers (APW)*: 1

Click the **Submit Changes** button to save the settings and start the database server. The following screen will appear:



Progress® OpenEdge® Management

Hello, admin

Dashboard Resources Alerts (74) Library Reports Jobs Database Administration

Resources / Local / myCDC

AdminServer: pcptu3

Database: myCDC

STOP DATABASE **STOP AGENT**

AdminServer: pcptu3

Database: Running

Monitoring Agent: Running

Active port(s): 1005

Configuration name: defaultConfiguration

Active page writers: Running: 1

After image writer: N/A

Before image writer: N/A

Watch dog: 12568

Database name: D:\WrkOpenEdge117\dbCDC\mycdc

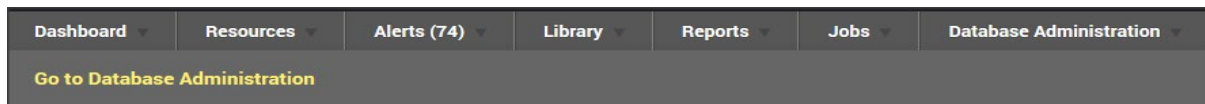
Passed (2 Minutes)

Last Poll: 2018-01-27 18:38:14

Poll Count: 2 Fail Count: 0 (0.0%)

[My Collections.Home](#)

Now we move on to the actual CDC configuration. In the main menu, click **Database Administration** then select **Go to Database Administration**.



In the list of databases, we select the myCDC database.

Database Administration			
Database Administration			
Connection Name	Database Name	Database Type	AdminServer
pcptu3.FathomTrendData...	fathom	Managed	localhost
pcptu3.myCDC	mycdc	Managed	localhost

A screen with a few views will appear. On the right side is the view of Database Features. Change Data Capture Features is off.

Database Features		
Review and enable database features		
Feature Name	Stat...	Action
Table Partitioning	✖	Enable
Read-only Partitions	✖	Enable
New VST Tables	✔	
Multi-tenancy	✖	Enable
Large Keys	✔	
Change Data Capture	✖	Enable

We will select **the Enable** box.

Database Administration / myCDC / Enable Change Data Capture

Enable Change Data Capture

This database is not enabled for change data capture.

To enable this feature, verify that you have the required database license for change data capture and provide the data area, index area details; then click Enable Change Data Capture.

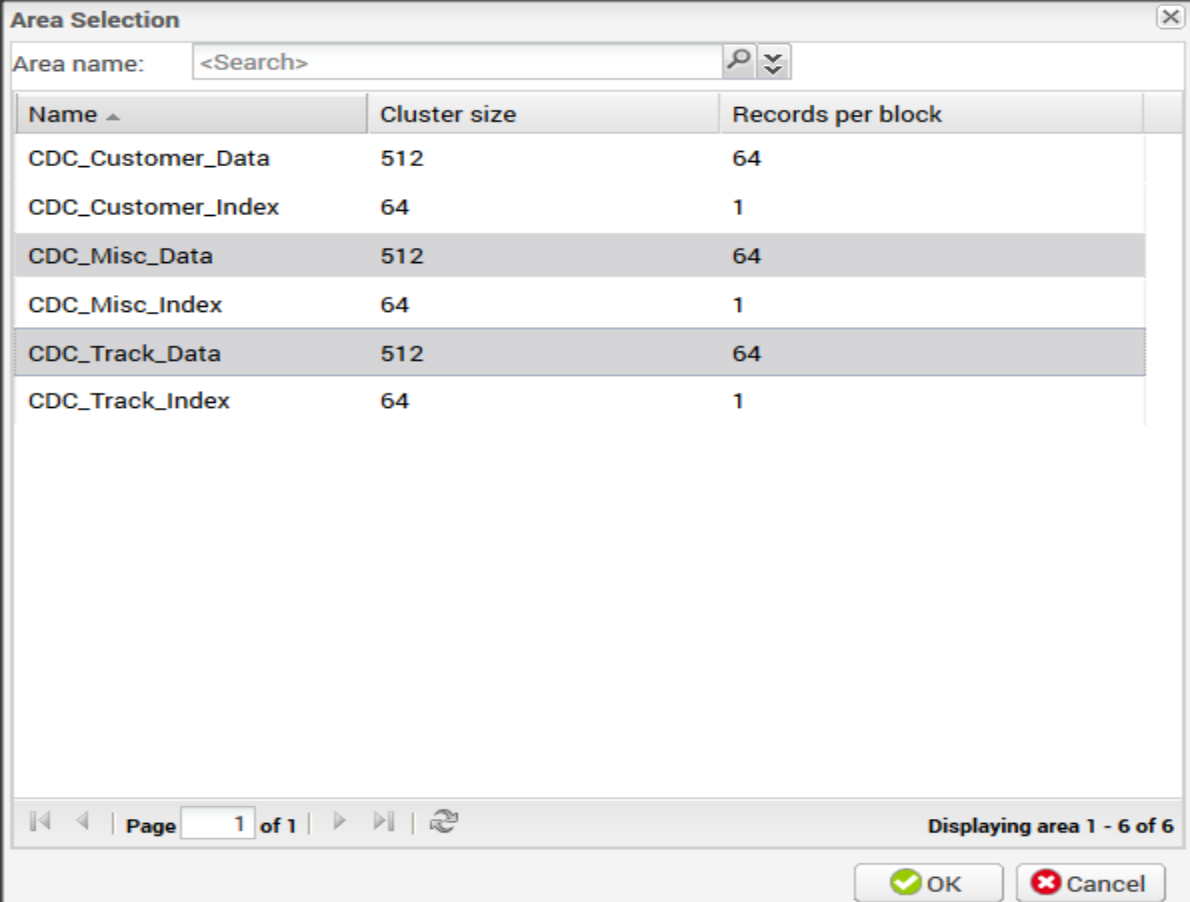
Setting the "Deactivate Index" as checked will create the change tracking table with inactive indexes.

Data area:

Index area:

Deactivate Index: ☐

Now we need to determine what areas the data and indices of the CDC will be stored in. Gradually click on the magnifying glass icon and select areas for storing data and indices.



Area Selection

Area name:  

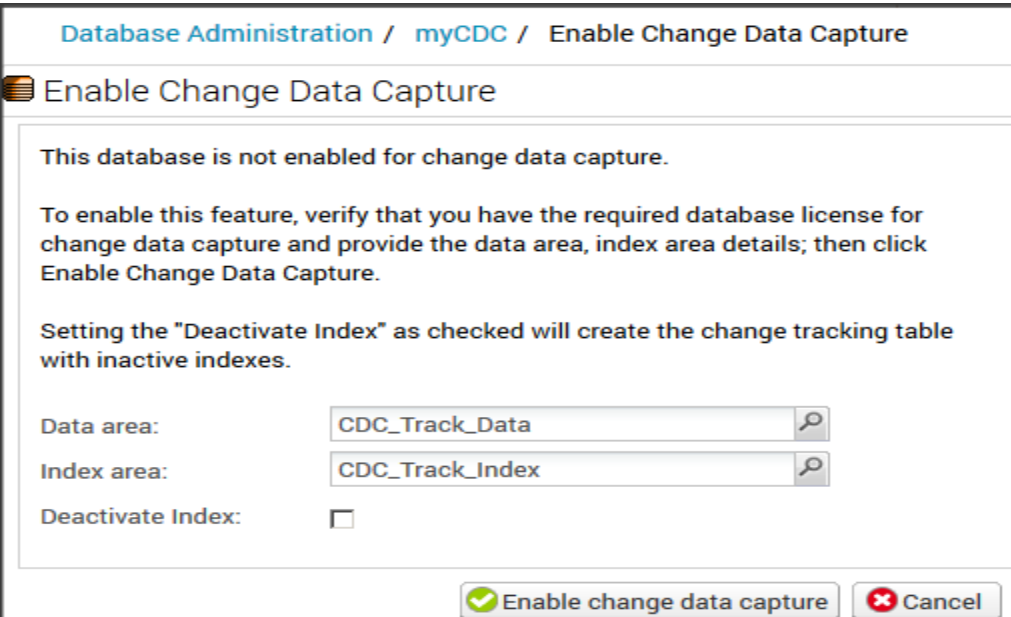
Name ▲	Cluster size	Records per block
CDC_Customer_Data	512	64
CDC_Customer_Index	64	1
CDC_Misc_Data	512	64
CDC_Misc_Index	64	1
CDC_Track_Data	512	64
CDC_Track_Index	64	1

Page of 1 

Displaying area 1 - 6 of 6

With the **Enable change data capture** button, we run the CDC.



Database Administration / myCDC / Enable Change Data Capture

 **Enable Change Data Capture**

This database is not enabled for change data capture.

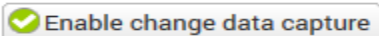
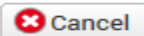
To enable this feature, verify that you have the required database license for change data capture and provide the data area, index area details; then click Enable Change Data Capture.

Setting the "Deactivate Index" as checked will create the change tracking table with inactive indexes.

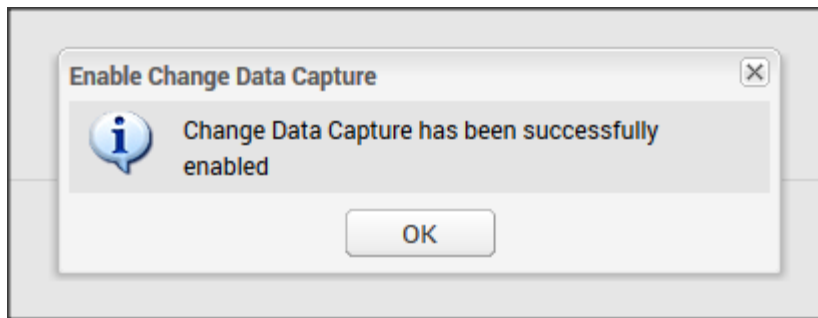
Data area: 

Index area: 

Deactivate Index: ☐

The result will appear in the following window:



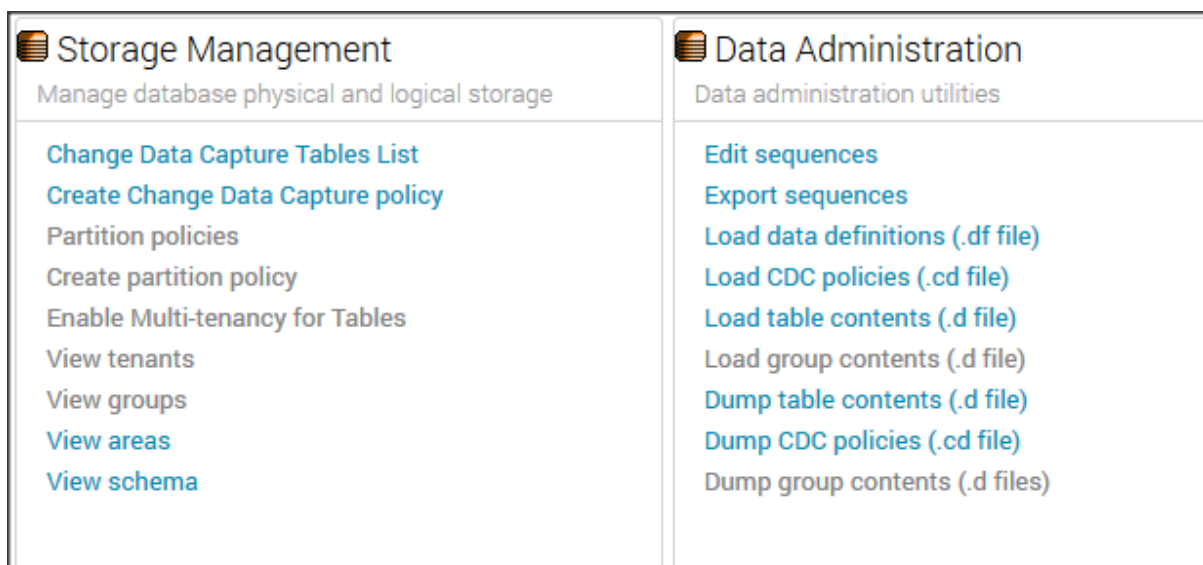
As experienced Progress users, we can also enable CDC directly from the command line. For example:

wicker myCDC -C enablecdc area Track_CDC_Data indexarea CDC_Track_Index

The database is now ready for CDC rule definition. In the following section, we'll show you how to set up these rules and introduce a few simple programs that demonstrate how to work with data from the Change Tables.

At the moment, we have a myCDC database with the CDC enabled function and defined areas for data and CDC indexes.

To configure the recording of changes, we will use OE Explorer with **the Database option Administration**.



In the left column, Storage Management list the options in which we can edit the list of tables or create/edit CDC (CDC policy) rules.

In the right section of the Data Administration are options for exporting and importing CDC rules.

When creating CDC rules, we need to specify the level (level) that determines the amount of data to be stored. It is illustrated in the following table:

Level	Description	Fieldmap	Possibility of change
0	Data storage only in Tracking Table.	NA	No
1	Data storage only in Tracking Table. Includes Fieldmap.	Contains only field information after update	Yes
2	Post-Update Values for All Operations (Create, Update, Delete)	Contains only field information after update	Yes
3	Write values before and after update for all operations (Create, Update, Delete)	Contains only field information after update	Yes

Fieldmap is used when we're only interested in identifying which fields have changed, without needing to capture their current values.

Now we're going to create a level 2 CDC rule.

In the Storage Management column, we select **Create Change Data Capture policy**.

Enter the following:

Policy name	CustomerPolice
Table	PUB.Customer
Level level	Medium (2)

We leave the State Pole as Inactive, we activate the rule later.

Database Administration / myCDC / Create Change Data Capture policy

CANCEL SUBMIT GENERATE POLICY PROGRAM

Create Change Data Capture policy

Policy name:

CustomerPolicy

Description:

Database connection:

myCDC

Table:

PUB.Customer

Level:

☐ Minimal(0)
☐ Minimal with Fieldmap(1)
☒ Medium(2)

State:

☐ Active
☒ Inactive

Encrypt policy:

☒

Identifying fields:

☒

Data area:

CDC_Track_Data

Index area:

CDC_Track_Index

Change table:

Change table owner:

In the Data area and Index area, we enter the names of defined areas that we created in the first article.

If we don't type a Change table, it'll have a default name with **CDC_** prefix, in our case **CDC_Customer**. The value of the Change table owner field will be set to **PUB**.

The choice of individual fields (columns) whose changes will be recorded is optional. If we want to select the fields of which changes will be recorded, we will select those fields.

Select from the list of fields by ticking the box in **the** left column of **the City, Country, CustNum, Name** field. For **CustNum** field select YES-fiction in the Enable field column, and in the Field order Add 1. An index will be defined in this field in the Change Table.

Select Change Data Capture Fields				
		Show All fields ▼	Note, Marking the checkbox on the field as checked v	
☐	Field name	Data type	Enable identifying field	Field order
☐	Address	character	☐ NO	
☐	Address2	character	☐ NO	
☐	Balance	decimal	☐ NO	
☒	City	character	☐ NO	
☐	Comments	character	☐ NO	
☐	Contact	character	☐ NO	
☒	Country	character	☐ NO	
☐	CreditLimit	decimal	☐ NO	
☒	CustNum	integer	YES	1
☐	Discount	integer	☐ NO	
☐	EmailAddress	character	☐ NO	
☐	Fax	character	☐ NO	
☒	Name	character	☐ NO	
☐	Phone	character	☐ NO	

By selecting the SUBMIT button (at the top of the page), we create a CDC rule that appears in the following window.

Database Administration / myCDC / Change Data Capture Tables list / Change Data Capture Table Summary

ADD CHANGE DATA CAPTURE POLICY

Change Data Capture Table Summary

Table name: Customer Table owner: PUB

X DELETE

☐	Policy Name	Level	State	Instance	Last Modified
☐	CustomerPolicy	Medium(2)	Inactive	Pending	Feb 12, 2018 10:01:01 PM

Navigation: 1

Select **Database Administration** in the main menu and **Go to Database Administration**, We can view the CDC Tables list.

Database Administration / myCDC / Change Data Capture Tables List

ADD CHANGE DATA CAPTURE POLICY

Change Data Capture Tables List

ACTIVATE DEACTIVATE Search Policy/Table SEARCH

☐	Owner	Table name	Change Table	Current Policy		Pending policy	Previous policy ...
				Policy name	Policy state		
☐	PUB	Customer		No Current Policy	No Current Policy	CustomerPolicy ⓘ	0

Navigation: 1 Displaying tables 1 - 1 of 1

The window shows the CDC rules for our **Customer** table. Policy state column shows **No Current policy** because we haven't activated this policy yet. We'll click the **CustomerPolicy** value in the Pending policy column.

Database Administration / myCDC / Edit Change Data Capture policy

CANCEL SUBMIT GENERATE POLICY PROGRAM

Create Change Data Capture policy

Policy name: CustomerPolicy

Description:

Database connection: myCDC

Table: PUB.Customer

Level: ☐ Minimal(0) ☐ Minimal with Fieldmap(1) ☒ Medium(2)

State: ☒ Active ☐ Inactive

To activate the rule, set the **State** switch to **Active** and click **Submit**.

Change Data Capture Table Summary

Table name:

Customer

Table owner:

PUB

×

DELETE

☐	Policy Name	▼	Level	State	Instance	Last Modified
	CustomerPolicy		Medium(2)	Active	Current	Feb 13, 2018 2:56:54 PM

⏮

⏪

1

⏩

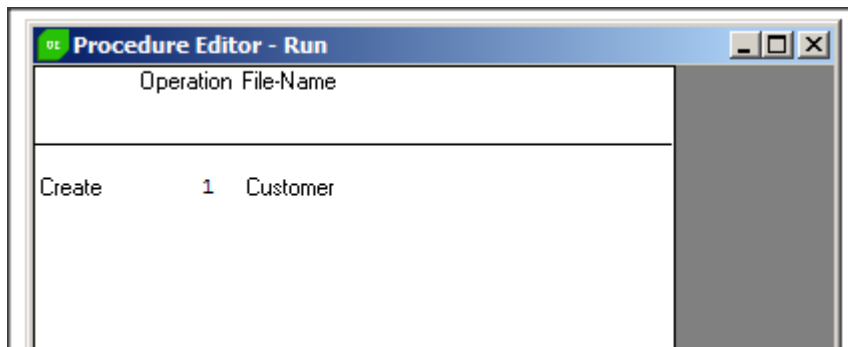
⏭

The **CustomerPolicy** rule is now activated. While it's possible to activate the rule at the time of its definition, we've done so here using standard ABL practices. Next, we'll demonstrate how Change Data Capture (CDC) works in practice and how to monitor changes in the relevant tables.

We'll start by creating a new customer using a simple ABL program with the **CREATE Customer** command and inserting a few basic values.

We'll view the **Change Tracking** records using the **Customer_CTT.p** program, which reads from the corresponding Change Table.

```
// Customer_CTT.p
FOR EACH _file, _cdc-change-tracking WHERE _file._file-number =
    _cdc-change-tracking._source-table-number AND _file._file-name="customer":
CASE _operation:
WHEN 1 THEN
    DISPLAY "Create" _operation _file. _FILE-NAME.
WHEN 2 THEN
    DISPLAY "Delete" _operation _file. _FILE-NAME.
WHEN 3 THEN
    DISPLAY "Before Update" _operation _file. _FILE-NAME.
WHEN 4 THEN
    DISPLAY "After Update" _operation _file. _FILE-NAME.
END CASE.
END.
```

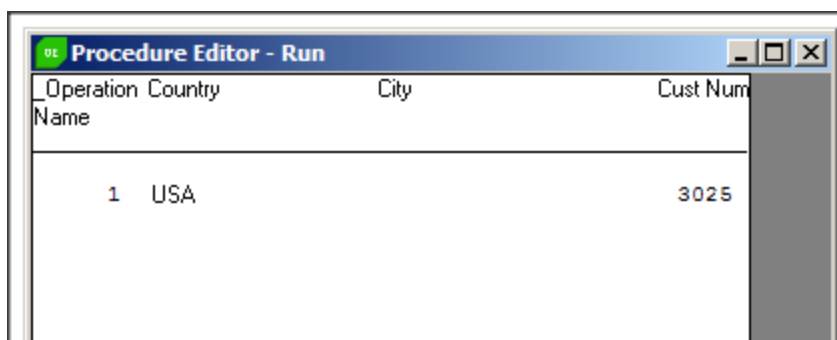


We can see that **Operation 1** (Create) has been recorded in the Change Tracking table. It's important to note that this table logs changes for all tracked tables, and the type of operation is stored as a number from **1 to 4**.

The **Customer_CT.p** procedure displays data directly from the Change Table **CDC_Customer**. This allows us to view the actual changed data—specifically, only the new values.

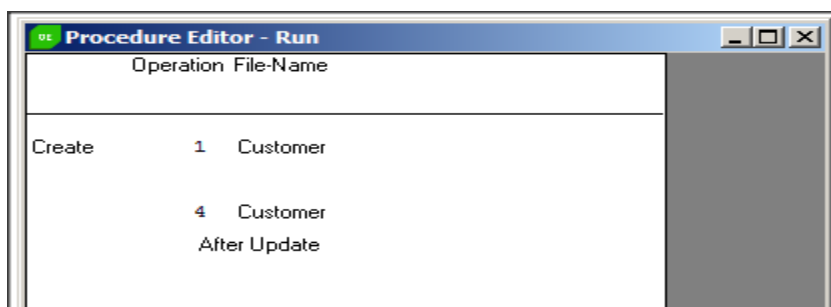
We'll view the Change Tracking records using the **Customer_CTT.p** procedure, which reads data from the **Change Tracking table**.

```
FOR EACH CDC_customer :
  DISPLAY _operation country city custNum name.
end.
```



Note that the Change Table copies only the columns from the original table that are configured to be tracked. It stores the new values for those fields when changes occur. Automatically generated values, such as CustNum, are also recorded.

Now, with a simple ABL program, we change the name of any customer from **Customer table**. Output from **Customer_CTT.p** procedure will look like this:



And from **the Customer_CT.p** procedure as follows:

Operation	Country	City	Cust Num
1	USA		3025
4	?	?	1

Lift Tours Corp.

We see a new value of **the Name** field and **the value of the CustNum** (this field is part of the index). The values of the other fields have not changed and are they displayed as ? (unknown value).