What is ETL?

Extract Transform Load (ETL) is a common terminology used in data warehousing which stands for extracting data from source systems, transforming the data according to the business rules and loading to the target data warehouse. In the early days of data warehousing this process was done by writing complex codes which with the advent of efficient tools we discovered was an inefficient way to process large volumes of complex data in a timely manner.

Why should we look at a different approach?

With the ETL tools it is true that there is less hand coding, easy maintenance and support, better metadata and data lineage, efficient impact analysis etc as features which make the life of ETL developers easy. But the processing power of the tool itself won't be utilized if the ETL process is well designed.

Generation one data warehouses were refreshed mostly once a month and as business intelligence became a 'must need' for organizations the demand to get data refreshed more frequently so that business can make decisions based on more current data become increasingly a necessity for current generation data warehouses. At one end of the spectrum is the monthly refresh and at the other is the near real time refresh. So the challenge most organizations which had an existing data warehouse faced was how and when to migrate to ETL tools. Well the industry has come through this and most of the industry has embraced ETL tools now. Next question is as the business demands increases how can we move from weekly to daily and per hour refresh. Definitely some of the traditional approaches will not work if the ETL has to be efficient. An efficient ETL process should be flexible enough to support the BI needs of an organization as the situation demands. The IT managers hence are pressured to make data warehouses available 24x7 with the most recent data at a low cost.

So what do we do?

One solution that seems to be working well is making the entire ETL incremental refresh based on new or changed data in the source systems.

The key for incremental process is identifying what data is net new or changed from last ETL process. Various methods like data comparison, audit columns etc were used to identify changed data. For large volumes of data this process is very inefficient. A better approach would be to use CRC (Cyclic Redundancy Check) approach which most of the ETL tools have built in. This is a welcome approach for any ETL where the option of CDC (Change Data Capture) that we are about to discuss is not an option.

Change Data Capture

CDC is an approach to capture and extract changes from operational systems. Most of the CDC products use a log scraping method to understand and keep track of the changes that happened to the source system. The advantages of using CDC approach are:

Minimal impact to operational systems as during ETL process only the log will be used and the database itself can be fully dedicated to the operational systems. This alone will make all the folks supporting operational systems very happy.

Data warehouse process and operational systems are completely decoupled from each other. This means any operational failure or recovery in either of these systems will not affect one another. This way the ETL can take its own time resolving a production problem and restarting without having to hold the operational systems. Since CDC uses only logs the changes happening to source system is captured and whenever the ETL process recovers it will have all the source changes till that point of time.

Only low volume of data is dealt by the entire ETL process and hence the process time is low. This reduction in ETL will enable multiple ETL runs during the batch window.

Extract could be scheduled during batch windows (pull) or near real time based on event/data or message based trigger (push).

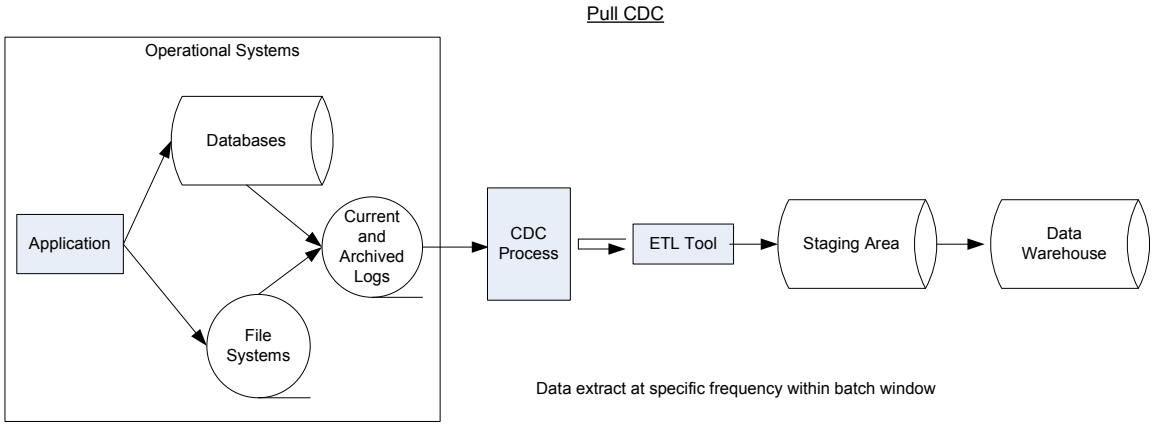
Along with the ETL tool vendors most of the major database vendors have started packaging CDC tools along with the database. Major players are IBM, Oracle, Datastage, Informatica and Attunity. Surprisingly SQL 2005 does not have a built in CDC application and I am sure Microsoft is listening to the BI industry and this will be available in future SQL server releases.

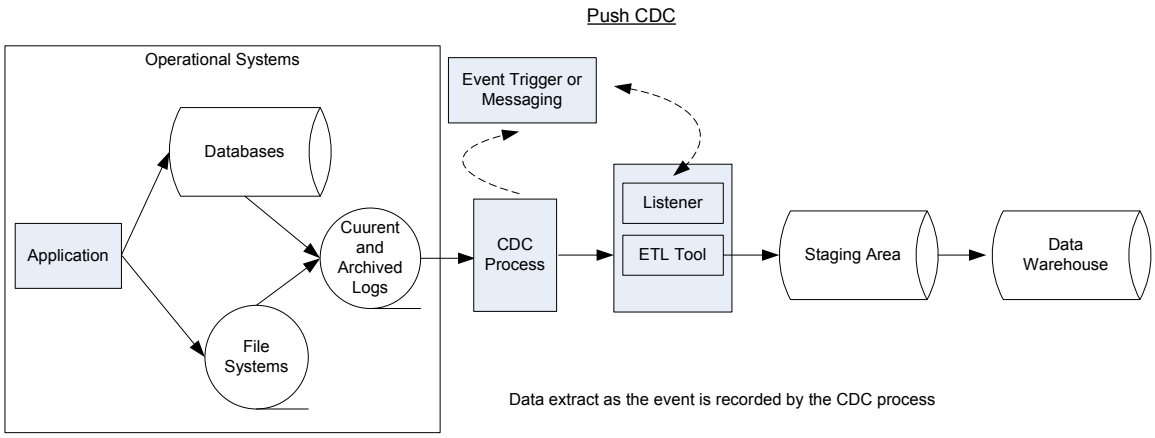
Pull CDC

ETL tool will extract data from the CDC process at predefined intervals. Only changed data since the last extract time is extracted.

Push CDC

An event trigger or messaging mechanism will notify the listener of the ETL tool about new data as soon as CDC process registers the changed data. The ETL tool will then extract the data to the data warehouse.





What are the things to watch out for ?

What happens when a new column is added into an existing table in the source system or if a source file layout is changed. Following steps are need in the above scenario
- Freeze data capture process
- Extract data to the data warehouse
- Make changes in the source database and the target data warehouse
- Start data capture process
- Run regular ETL process

Committed or uncommitted transactions?

Based on the business requirement (how the data should be in the data warehouse) decide whether only committed transactions should be extracted or dirt ready is acceptable.

Does the business need to see all the transactions/events that happened on a row or is it enough to see the last status before the ETL process?

Based on business requirement decide if to extract the last known status or to show all the events if there were tons of action on a row. This to an extent is also based on type of dimensions used. If type 2 or 3 dimensions exist in the data warehouse you almost for sure want to see all the events except if in situations where your business says what constitutes a type 2 is the changes for the same row between the statuses at the end of day. This can be set in the CDC process and is very important to think about it during your ETL design as the number of rows that flow to the ETL will be impacted based on the decision. For example if the same data row can change multiple times during the course of the day and if the data warehouse needs only the last status before the ETL kicks in during the batch then CD process should be set to give only 1 row. On the other hand if there is need to see all the statuses then CD should be set to give multiple rows for the same data representing each event. The impact for ETL is the process time difference between 1 row and 5 rows and when we look at then thousand transactions in the table the above decision will make a good impact on the ETL time.

How will the ETL determine what to process and what not to look into?

Mind you the CD process is running continuously, it is constantly recording changes that happen in the source system. If you have a 24x7 business and if your data warehouse SLA is also 24x7 the question is when do we extract the data? The approach is to let CDC process know what timeframe of data you are interested in. If the ETL starts at 7:00 pm and the last successful extract was at 6:00pm then let ETL process must extract data between 6:00 pm and 7:00 pm. This is usually done by looking at the internal log timestamp which the CDC process had captured from the log for each transaction. In the above example 7:00pm is the cut off time and 6:00 pm is start time. In CDC process that maintains a database table the ETL process can always updated the consumed rows with a flag which is useful for later audit purpose. This may not be always possible and depends of the CDC application.

In the above example the question is what happens to a transaction that started at 6:59:59. Well it depends on as indicated earlier what does the business want to see, just committed transactions or they don't mind seeing uncommitted transactions. Based on what the business wants change appropriate setting in the CDC application. For those who are concerned what happens if an uncommitted transaction is extracted to the data warehouse and what if the data is changed and committed at the end of the very transaction. The answer is till the next ETL extract the data warehouse will show the uncommitted data and during next ETL the data warehouse will update the data.

How does the typical CDC data look like? There will be a minimum of the following fields in the CDC application
Table name
Data columns
Change Indicator (For Example 'I' for Insert, 'U' for Update, 'D' Delta etc)
Log time