

The Complete Guide To Data Replication

MOVE DATA AT THE SPEED OF BUSINESS WITH
CHANGE DATA CAPTURE

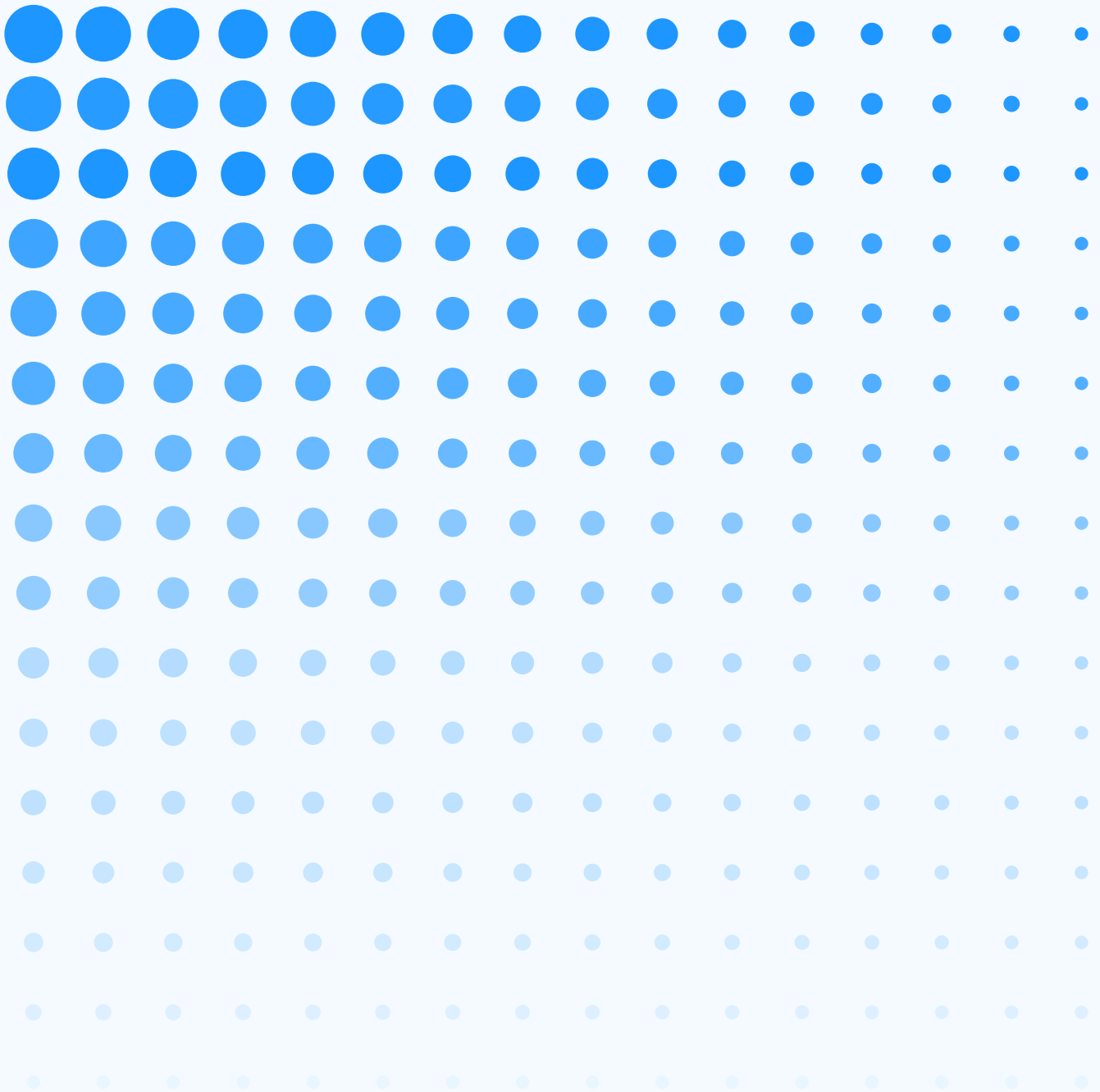
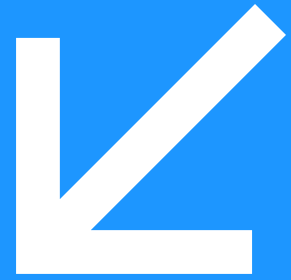


Table Of Contents



INTRODUCTION	1
WHAT IS CHANGE DATA CAPTURE?	2
THE HISTORY OF CHANGE DATA CAPTURE	3
4 TYPES OF CHANGE DATA CAPTURE	4
THE BENEFITS OF CHANGE DATA CAPTURE	5
THE USE CASES OF CHANGE DATA CAPTURE	6
CHANGE DATA CAPTURE AND ETL	7
HOW TO IMPLEMENT CHANGE DATA CAPTURE	8
CONCLUSION	9

01

INTRODUCTION

Forward-thinking companies understand the transformational power of data for their business.

Research by McKinsey & Company has shown that “data-driven” organizations enjoy a wide range of benefits, from retaining and satisfying more customers to achieving greater profitability.

It’s not just collecting this information, however, that sets data-driven businesses apart from the rest—you also need to efficiently process and analyze it to uncover hidden trends and discoveries. Yet the vast majority of companies accumulate far too much data on a daily or even hourly basis to be manually inspected by human employees.

Rather, organizations need robust, efficient ways of moving this information through their data pipelines so they can enjoy fresh, up-to-the-minute insights and ideas.

One such data integration method is known as change data capture (CDC). So what is change data capture, exactly?

Why are businesses large and small becoming increasingly interested in CDC? What are the different benefits and applications of CDC, and how does CDC align with other data-driven processes such as ETL? In this white paper, we’ll offer everything you need to know about CDC as we answer all of these questions and more. Protected health information (PHI)



02

WHAT IS CHANGE DATA CAPTURE?

During the data integration process, most organizations extract information from one or more sources and collect it in a centralized repository (a process known as ETL, which we'll discuss in detail later).

But how do you know when your sources actually have new data for you to capture and integrate? If you perform ETL too frequently, you'll simply be wasting computing power and effort collecting the same data over again. If you perform ETL too infrequently, however, you could be missing out on vital new information, impairing your decision-making abilities.

Striking the right balance is especially crucial because much of your enterprise data is “perishable”—i.e. its value diminishes rapidly over time. Failing to promptly intake this time-sensitive information will directly affect your ability to become a data-driven organization, beating your competitors and better serving your customers.

That's exactly where change data capture comes in. Change data capture (CDC) is a term for the collection of processes and techniques that detect when changes have been made to a source table or source database. CDC techniques identify the information that has been added or updated at the source since the previous execution of ETL. For maximum speed and efficiency, you can then limit your ETL workloads to only these changed rows or entries, collecting them in a target location with the rest of your enterprise data and using them for business intelligence and analytics.

03

THE HISTORY OF CHANGE DATA CAPTURE

Change data capture has been gaining a great deal of attention these days, but it hasn't simply emerged out of the blue. In this section, we'll discuss three larger trends in business IT that have prompted the rise of CDC.

1. The growth of big data

With every day, hour, or minute that passes, data-driven organizations continue to intake new information—whether that's in the form of database tables, system logs, websites, Internet of Things (IoT) devices, or several other possible data sources. These data-driven organizations know that information is one of their most valuable assets, and devote a great deal of their IT operations to efficiently storing, processing, and analyzing this information.

Yet the time has long passed when these businesses could manually plumb their information for insights themselves. [According to a 2016 study by IDC](#), the average company manages 163 terabytes, or 163,000 gigabytes, of data (and by now, the figure is likely substantially higher).

As new information arrives by the second, organizations need a powerful, automated method of examining this data and extracting the most useful bits and bytes—which is exactly what change data capture intends to do.

2. Migrating to the cloud

Cloud computing has become a business best practice for companies the world over, achieving wide-scale popularity. According to Flexera's [2021 "State of the Cloud" report](#), 59 percent of companies call themselves "heavy" users of cloud computing, with an advanced level of cloud maturity.

In particular, cloud migrations have contributed to the rise of change data capture. Moving your data storage, processing, and analysis to the cloud unlocks the potential for better performance and smarter data-driven insights. As we'll discuss in the next section, cloud migrations have given businesses unprecedented IT agility, flexibility, and scalability—enabling the rise of real-time data processing which is a major motivation for CDC.

3. Batch processing to streaming

Traditionally, ETL jobs have been run in batches at regularly scheduled intervals. For example, you might run a labor-intensive ETL job overnight, so that the insights you extract are available on the desks of managers and executives the next morning. However,

technological advances and the fast-paced business landscape have together led to demands for increasingly frequent ETL workloads, and even real-time ETL (also known as streaming ETL).

Real-time ETL is a must for use cases such as fraud detection. Businesses may handle thousands or millions of financial transactions on a daily basis, and for each one, they need to pull and analyze the relevant information to determine whether a given event is fraudulent. Data sources such as social media and IoT devices, which are constantly producing new information, also lend themselves well to streaming ETL.

Of course, real-time ETL simply isn't possible without a functional solution for change data capture. (Otherwise, businesses would have to load all of their data, all over again, every second of every day.) This has led to a growth of interest in CDC as one of the essential underlying technologies of streaming ETL.

04

4 TYPES OF CHANGE DATA CAPTURE

So far, we've briefly defined change data capture and gone over the various motivations that have led to the rise in interest for CDC. In this section, we'll discuss the four major types of change data capture, each of which has its own advantages and disadvantages.

1. Log-based CDC

In log-based CDC, the change data capture solution analyzes a database's metadata file known as the transaction log. This file records all of the changes, in order, that have been made to the database (such as INSERT, UPDATE, and DELETE statements).

Although the transaction log is primarily intended to help the database recover from an unexpected crash, it can also be used by CDC solutions to reconstruct the changes made to a database and its tables since the previous execution of an ETL process. The CDC technology can then replicate these changes to the target centralized repository (e.g. a data warehouse or data lake).

Database transaction logs are highly reliable, up-to-date, and accurate, which means that log-based CDC also tends to be accurate and reliable. Another benefit of log-based CDC is that the transaction log is separate from (and smaller than) the database itself, making it fast and simple to identify the relevant changes without unintentional side effects.

However, log-based CDC also comes with its own downsides. Most notably, each database technology (e.g. MySQL, Microsoft SQL Server, Oracle, etc.) may come with its own proprietary format for its transaction logs. This means that you may have to create a custom CDC solution for each type of database that you want to pull from. What's more, not all types of databases or data sources may even maintain a transaction log, making log-based CDC impossible in these cases.

2. Trigger-based CDC

Instead of transaction logs, trigger-based CDC uses special functions known as database triggers, which run when another event occurs. For example, the BEFORE UPDATE trigger in MySQL occurs prior to an update to a database table. The CDC solution can detect when one of these trigger functions executes, and then update the target repository accordingly.

Like log-based CDC, trigger-based CDC is relatively simple to implement because it depends on the database's functionality itself (transaction logs and/or database triggers).

For some databases, trigger-based CDC even receives direct support in the SQL API (application programming interface).

Trigger-based CDC also comes with a performance tradeoff: it entails less overhead when extracting database changes, but more overhead to the operation of the database (since the triggers are extra functions that need to run). Finally, triggers may be disabled during certain operations, and some databases or data sources may not even come with trigger-based CDC enabled.

3. Metadata

A third option for change data capture is to directly examine the metadata of the database or data source. Many databases, for example, have a column such as `DATE_MODIFIED` or `LAST_UPDATED` that denotes when a particular row or entry has been changed. By scanning these columns and filtering those records that have changed since the most recent data extraction, you have an effective CDC method at your fingertips.

The greatest benefit of this metadata approach is that it's simple and easy to implement.

However, writing a custom script is also brittle and can stop working if the underlying database schema changes. Perhaps more importantly, the metadata approach also may not work if you want to keep track of deleted rows or entries. In this case, since the deleted records are no longer present in the source database, there's no easy way to determine that they were ever-present at all.

4. DIFF

If the lack of detection for deleted records concerns you, you can also try a direct comparison between the current database and the database state at the time of the previous ETL workload. The good news is that this is easily implemented with a terminal command such as `diff` in Unix-based operating systems (also known as `fc` in Windows).

The disadvantage of the diff approach is that it's slow and resource-intensive. Not only do you have to keep around two copies of the same database (the current and the previous), you also have to do a line-by-line comparison between them, which can take a great deal of time. Being slow and thorough with the diff CDC method is the price you have to pay to be able to identify deleted records.

05

THE BENEFITS OF CHANGE DATA CAPTURE

Change data capture offers a wide variety of benefits for data-driven organizations. The advantages of implementing CDC include:

Real-time decision-making: One of the biggest benefits of CDC is that it enables faster, even real-time decision-making. For companies that prioritize speed, agility, and responsiveness, CDC can be a make-or-break factor that enables them to capitalize on streaming data sources. Retail businesses, for example, can use CDC to provide shoppers with personalized experiences and offers every time they walk into a store, based on their previous interactions with the business.

Savings on time, money, and effort: CDC is an automated process, which means that your human employees don't have to waste their valuable time and effort on tedious manual data integration. A well-oiled CDC solution can continuously feed your BI and analytics workflows with fresh data without disrupting your employees or the rest of your IT environment.

User-friendliness: By elegantly automating and concealing the process of updating your databases, CDC makes the data integration process more user-friendly. This is ideal for organizations with fewer technically skilled employees, who may not have the knowledge or capacity to create custom data integration scripts.

06

THE USE CASES OF CHANGE DATA CAPTURE

Perhaps the best-known and most widely used application of change data capture is for ETL. While we'll discuss CDC and ETL more in-depth later, this section will go over some of the most common CDC use cases.

1. Database synchronization

Change data capture is essential in order to keep two or more databases synchronized with each other. For example, centralized data repositories such as a data warehouse often serve as a “single source of the truth.” These databases store “golden records” which contain the most complete, accurate, and up-to-date information. However, your IT environment may also have secondary databases that need to be updated at regular intervals with any incremental changes to the master database.

In these cases, CDC is an invaluable tool for real-time synchronization between multiple databases, software, or systems. Once a record is changed inside the centralized master database, this change can then be detected by CDC and propagated to other databases throughout your environment.

2. Operational data stores

An operational data store (ODS), like a data warehouse, serves as a centralized repository for your enterprise data. Unlike a data warehouse, however, an ODS has been designated specifically to store information

regarding a specific practice, domain, or business area—for example, finance, sales, marketing, or customer support.

Just like a data warehouse, an ODS needs to be updated regularly so that users can enjoy up-to-the-minute information and insights, helping them in their daily business activities. With change data capture, organizations can efficiently and continuously make the necessary modifications to an ODS, rather than having to poll for changes at scheduled intervals.

3. IT Audits

Last but not least, change data capture solutions can be a lifesaver in the event of an IT audit. Depending on what auditors are looking for, you may be required to maintain a paper trail of database entries or transactions in order to clear up any suspicious activity.

But this is exactly what CDC solutions are intended to do: maintain a more or less permanent record of any changes or modifications to a database. As such, CDC logs can have an unintentional yet highly useful purpose: acting as a historical record of a database or table if auditors come knocking on your door.

07

CHANGE DATA CAPTURE AND ETL

Perhaps the best-known and most widely used application of change data capture is for ETL. While we'll discuss CDC and ETL more in-depth later, this section will go over some of the most common CDC use cases.

The three steps of the ETL process are as follows:

- Extracting data from one or more sources (e.g. relational and non-relational databases, flat files, websites, APIs, SaaS platforms, etc).
- Transforming the data in order to clean it and prepare it for the target repository.
- Loading the data into a centralized destination such as a data warehouse or data lake.

As you can guess, CDC is most effective during the first (i.e. "extraction") phase of ETL. With the right CDC solution, you can detect which records have changed (and which have not) from all of your ETL sources. This allows you to avoid duplicating information that has already entered your data warehouse. CDC can be effective either for traditional batch processing or for a more modern real-time streaming ETL workflow.

08

HOW TO IMPLEMENT CHANGE DATA CAPTURE

This white paper has addressed a wide range of topics: what CDC is, the various forms of CDC, the benefits and applications of CDC, and how CDC intersects with ETL in particular. There's just one important question left to answer: how can you actually implement change data capture within your organization?

First, there's no "one size fits all" answer to this question. The way to implement change data capture will heavily depend on the specifics of your IT environment and the data sources you work with.

Microsoft SQL Server databases, for example, use log-based CDC, which means that you'll need a CDC solution that can work specifically with SQL Server transaction logs. (We've written a separate article on change data capture in SQL Server that you can peruse [here](#).) Here's a quick overview of what you need to do for change data capture in SQL Server:

- CDC is available for all Windows users since SQL Server 2008. CDC is also available for Linux users starting with SQL Server 2019, or with SQL Server 2017 if Cumulative Update 18 (CU18) is installed.
- You must be running SQL Server Developer, Enterprise, or Standard Edition—the web edition of SQL Server does not support CDC.
- A SQL Server sysadmin must enable CDC with the procedure `sys.sp_cdc_enable_db` for an entire database, or the procedure `sys.sp_cdc_enable_table` for a single table.

The first five columns of a SQL Server change table contain metadata that is relevant for CDC. This includes the commit log sequence number (LSN) for a given change, the type of change (e.g. an insert, update, or delete operation), and the columns that were affected by the change.

As you can see, there are quite a few technical considerations that organizations need to account for when performing CDC for a specific database (such as SQL Server). Now consider the fact that you may use dozens of different data sources within a single ETL workflow, and the possibility of manually configuring CDC quickly begins to seem unmanageable.

Instead of setting up CDC for all of your data sources individually, most businesses think smarter and choose an ETL tool with built-in CDC functionality. By automatically handling the technical issues behind the scenes, these CDC-enabled ETL tools allow users to streamline and simplify the data integration process.

When looking for an ETL tool with change data capture, think about the following factors:

Diversity of sources: At minimum, you should choose an ETL tool that integrates well with your existing tech stack, and that can perform CDC on your existing databases and other sources. Also consider whether your needs may change over time and whether you require an ETL tool that can run in many different environments.

Performance: Speed and performance are always important when it comes to CDC and ETL, especially for real-time use cases. If these issues are a predominant concern for your organization, take the time to evaluate the underlying technologies and architectures of the ETL tools you're considering.

Ease of use: CDC-enabled ETL tools vary in terms of their user-friendliness and the technical knowledge required to use them. Selecting an ETL tool with a gentler learning curve can help encourage user adoption and lower the total cost of ownership.

09

CONCLUSION

Change data capture (CDC) is a catch-all term for a number of techniques that detect when underlying data records or databases have changed since they were last examined. CDC is a crucial methodology for use cases such as ETL, which is the predominant form of data integration for enterprise data.

So how can you leverage the power of change data capture for your own organization? More and more businesses are turning to dedicated ETL and data integration tools like Integrate.io.

Integrate.io is an ETL platform with CDC functionality that makes it simple to build automated pipelines between your data sources and your cloud data warehouse. The Integrate.io platform offers both log-based and trigger-based CDC, as well as other crucial features such as a no-code drag-and-drop interface and more than 100 pre-built integrations and connectors.

Thanks to Integrate.io, it's never been easier to construct powerful, robust ETL data pipelines with CDC functionality, offering your users fresh, accurate information at all times. Ready to learn more about how Integrate.io can help with ETL, data integration, and change data capture? [Get in touch with our team of data experts today](#) for a chat about your business needs and objectives, or to start your 14-day pilot of the Integrate.io platform.