ORACLE

Oracle GoldenGate Advantages

Comparison of Change Data Capture (CDC) techniques for the Oracle Database

August 2025

Public

Purpose statement

This document provides an overview of how Oracle GoldenGate (also known as GoldenGate) works, and why its architecture makes it the best overall choice for the Oracle Database Change Data Capture (CDC).

When choosing a technology for CDC and data replication, it is crucial to understand in depth how the data events are being captured and processed. Every database has one or more different options to detect data events – each different option comes with trade-offs that should be understood and carefully considered. For example, the Oracle Database has had as many as six different ways to detect data events.

In this document we will discuss technical differences between CDC approaches for Oracle that include database triggers, LogMiner API, XStream API and GoldenGate. We will also briefly discuss the risks associated with unsupported third-party log readers that attempt to reverse engineer the redo log record structures.

Disclaimer

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle. Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

Table of contents

Introduction: A Category Leader

Oracle GoldenGate has been long recognized by customers and analysts as the most functionally complete, highest performing, and most trusted data integration and database replication solution not only for Oracle, but non-Oracle platforms. As a proven heterogeneous solution, it integrates with 100's of combinations relational and non-relational databases, data stores and clouds. For the Oracle Database, GoldenGate is the only scalable, and fully supported solution with the broadest support for Oracle database features. Other frameworks and tools for capturing data events from Oracle Database have incomplete feature coverage, are non-scalable, and/or unsupported. For example, LogMiner is a single-threaded diagnostic API with limited capabilities for real time capture. Third-party data integration technology that depends on the LogMiner API cannot come close to matching Oracle GoldenGate for use with the Oracle Database.

The focus and purpose of this paper is to explain the options for capturing data events from the Oracle Database, how GoldenGate fits in, and why it is your best option.

Oracle GoldenGate

A San Francisco startup founded in the 1990's, GoldenGate's original purpose was to provide business continuity and high availability for networked ATM/cash machines running mission critical databases.

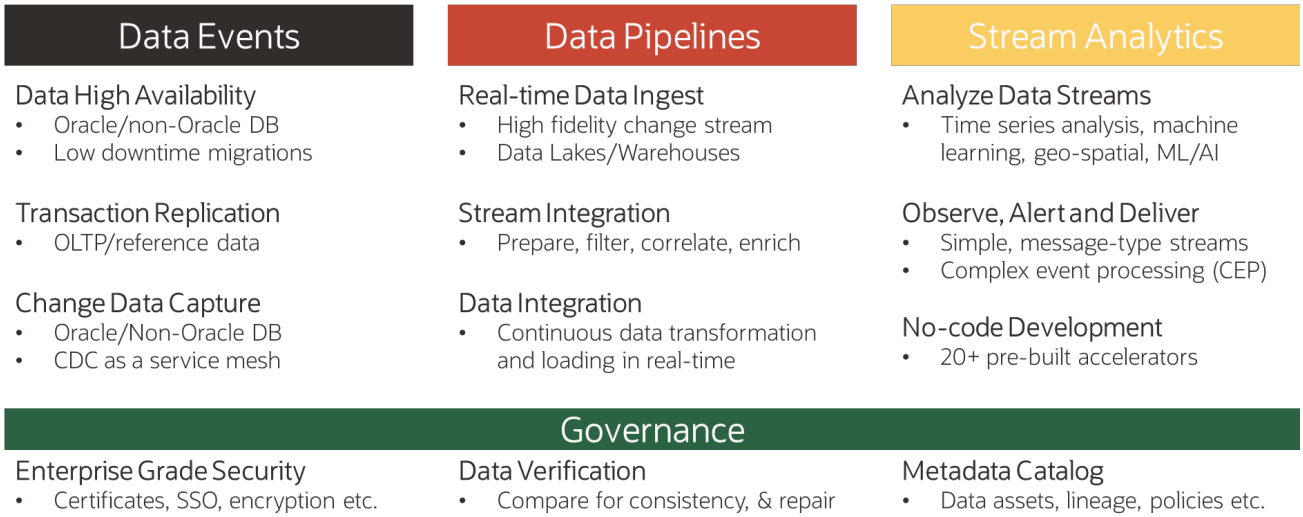

Figure 1: Oracle GoldenGate platform capabilities

Oracle GoldenGate is recognized as the leader in data integration and analytical capabilities serving a broad range of data sources and data targets. The reliability of exactly once, guaranteed transaction delivery has enabled GoldenGate to be the anchor technology to deliver Oracle's MAA 'Platinum Tier' service level in use on the most mission critical Oracle databases. This kind of reliability extends to non-Oracle databases such as PostgreSQL, MySQL, SQL Server, DB2 iSeries, mainframes and other supported data stores. Many thousands of global financial institutions, technology, retailers, telecoms, healthcare companies etc. run their operational data platforms on the trusted foundation of Oracle GoldenGate.

GoldenGate is a real-time data replication platform that can detect data events and route them across networks at very low latencies. The GoldenGate technology is used for geographic sharding of operational databases, low-

downtime data migrations, multi-active (online) data stores, real-time data ingestion to cloud, data lakes, and data warehouses etc. For more than a decade GoldenGate has been increasingly focused on polyglot big data and noSQL data payloads and has been completely refactored for native microservices 'as a service' deployments.

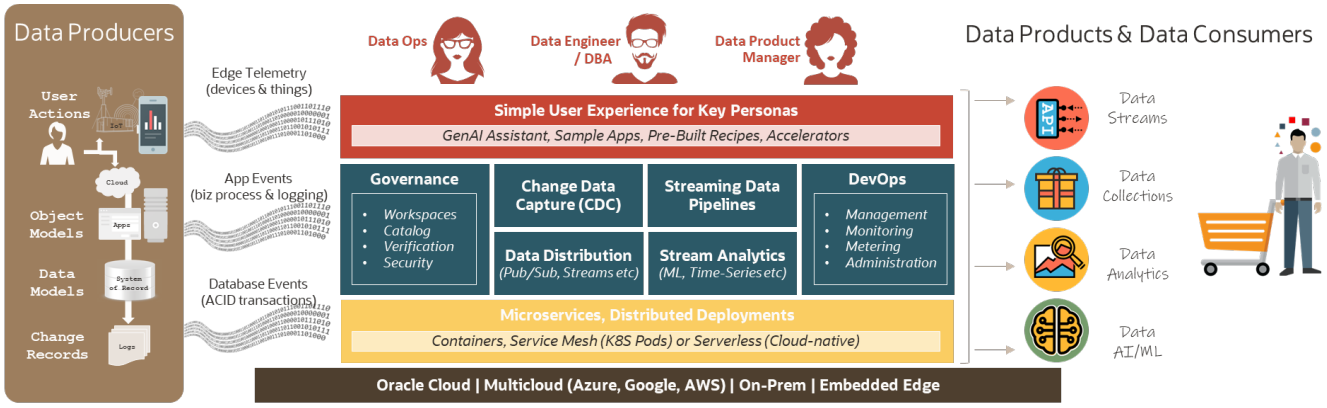

Figure 2: Oracle GoldenGate platform. Logical architecture and key components

In 2018 the GoldenGate platform added Data Pipelines and Stream Analytics with a robust complex event processing (CEP) core engine that scales to billions of events per second while preserving ordered data processing down to the nano-second scale. This event engine can use very powerful semantics for transformations or analytics and runs on an open-source Apache Spark for massively parallel processing (MPP).

With these new capabilities, GoldenGate provides high-value data products directly to data consumers. In the past, GoldenGate excelled at delivering low-latency raw data to data pipelines or other data products. These new features all GoldenGate to can push raw data events as well as provide high-value data products.

How GoldenGate Works

At heart, the GoldenGate CDC capabilities are used for detecting and transmitting data events. Data events typically include database DML (inserts, updates, deletes), DDL, and special database changes like an enqueue or dequeue operations, or sequence value change. On the Oracle Database, GoldenGate can also detect and transmit procedure events, which is useful when keeping databases in sync. GoldenGate has been designed from the ground up to be a trusted provider of data events and can be trusted for MAA (maximum availability architecture) use cases, as well as for DW (data warehouse), Data Lake and data streaming (e.g. Kafka-based) scenarios that are mission-critical.

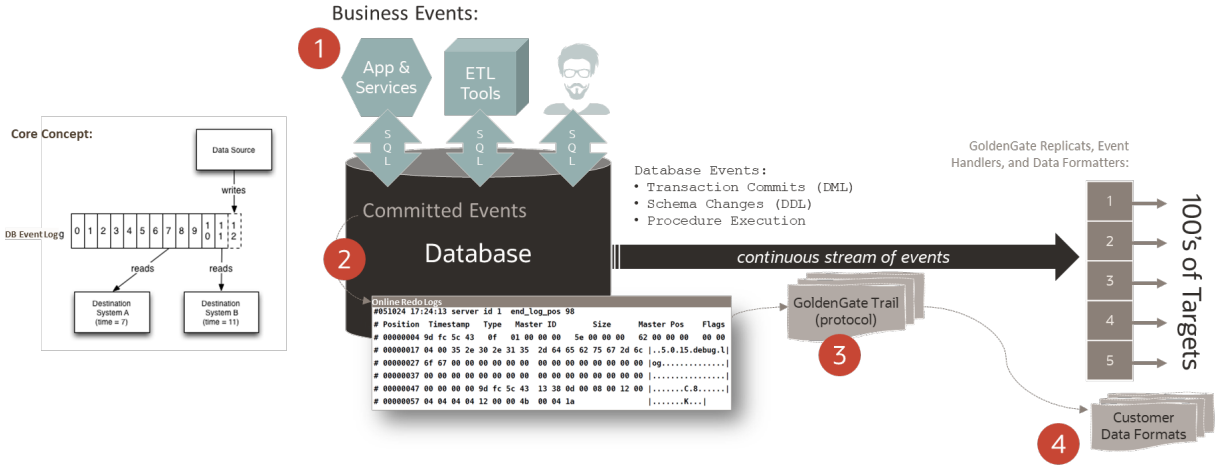

Figure 3: Change detection and propagation in Oracle GoldenGate

Anatomy of a GoldenGate Transaction

As database log events are captured, they are converted into the GoldenGate canonical ledger called a GoldenGate Trail. All the data event logs of the 100's of combinations of supported sources/platforms are converted into this single canonical Trail format. Each Trail may have one or more 'paths' to GoldenGate deployments on other servers, and Trail consumers may initiate their own access to a path, or GoldenGate can be configured to push Trails downstream.

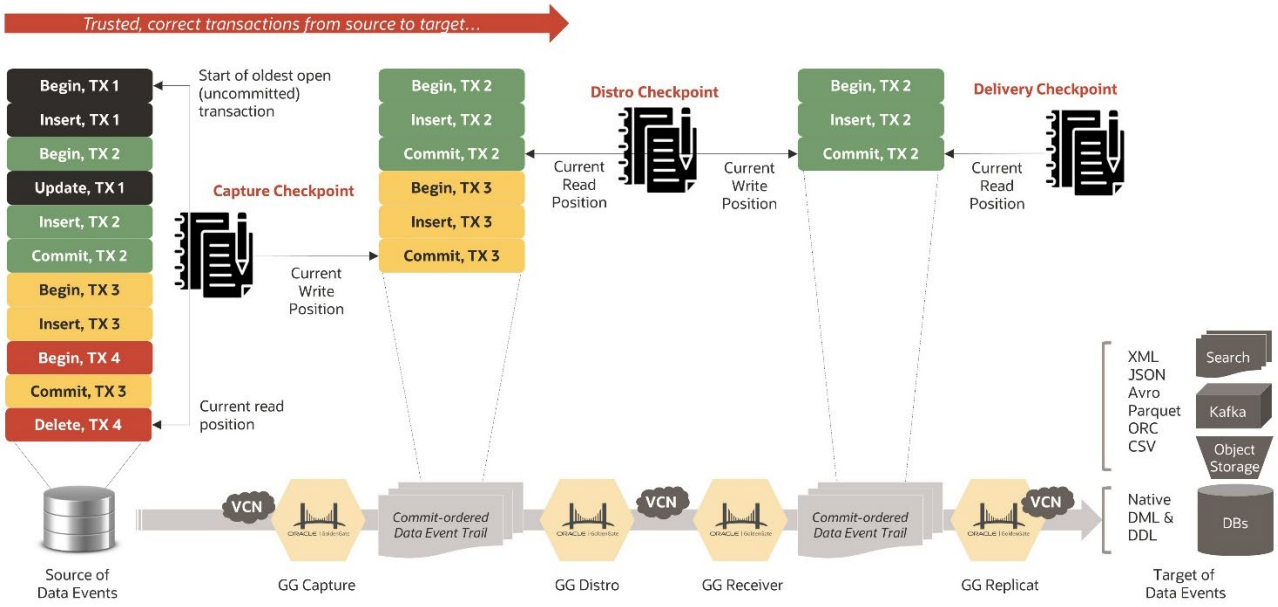

Figure 4: Anatomy of a GoldenGate transaction, preserving ACID properties in a distributed architecture

GoldenGate is aware of all source operations as they hit the database logs, and each different kind of database handles operations differently. As a rule, GoldenGate emits the data events once they are committed on the source system. GoldenGate groups together, isolates and preserves the consistency of these transactions for optimal performance as they move across the networks.

Options for Detecting Events in Oracle Database

Over the years, Oracle Database has had many APIs that can be leveraged for changed data capture (CDC). Today, there are handful of techniques that remain a strategic part of the Oracle ecosystem:

- **Oracle GoldenGate** is the best-performing and most feature-rich technology to detect changes and replicate data events from the Oracle Database. A modern microservices architecture makes GoldenGate an ideal solution for all your real-time data movement needs. (documentation)
- **Oracle XStream** consists of Oracle Database components and application programming interfaces (APIs) that enable client applications to receive data changes from an Oracle database and send data changes to an Oracle database. XStream is licensed with Oracle GoldenGate and is used by ISV application partners and non-Oracle replication tools to support high-speed CDC from the Oracle Database. (documentation)
- **Oracle LogMiner** is a diagnostic tool developed to troubleshoot issues related to the redo logs of an Oracle Database. This tool enables users to query online and archived redo log files through a SQL interface. LogMiner is a popular option with many ETL and Replication vendors because it is an API of the database that is free to use without any additional licensing. With LogMiner activated, there may be some noticeable degradation in the update performance of Oracle Database. (documentation)
- **Database Triggers** are a stored PL/SQL block, which may be associated with a Table and execute when DML statements such as INSERT, UPDATE, or DELETE occur. Using Triggers to detect events require a detailed

knowledge of the schema and application which is prone to employee turnover and poor documentation. This highly invasive strategy can be both incomplete and incur lots of runtime overhead. Nevertheless, many ETL vendors continue to support using CDC with Database Triggers. (documentation)

This document will also discuss one unsupported alternative for detecting changes in Oracle Database:

- **Unsupported Third-Party Disk Log Readers** use reverse-engineering and pattern matching to mine the Oracle Database redo logs and archive logs directly from the server storage without using a provided API. This approach is unsupported by Oracle, poses measurable security risks and may be in violation of Oracle license agreements. Later in this document we will explain the risks involved with this technique.

Oracle GoldenGate is the recommended approach for CDC and replication with Oracle Databases. It supports more than 300 combinations of sources and targets including all versions of Oracle Database at all patch levels. GoldenGate uses its own deeply integrated log capture API for detecting data events and reading log records from Oracle DBs, enabling the least invasive, fastest and most scalable solutions.

Changed Data using Oracle GoldenGate

Oracle GoldenGate leverages exclusive API calls inside the Oracle Kernal. This allows GoldenGate to take advantage of new database features as they become available. The API is deeply integrated with the database kernel and is optimized for high-speed processing of data events. Due to the tight integration of the API with the database kernel it allows for the broadest level of feature support and at the highest volumes while at the same time providing the least overhead of Oracle's CDC technologies. GoldenGate is the only replication framework that is certified for Oracle's Platinum Tier of the maximum availability architecture (MAA). (documentation)

As with the MAA technical paper noted above, GoldenGate is often run in a 'hub' configuration. This is advantageous for single point-projects and for simplifying the administration of GoldenGate components.

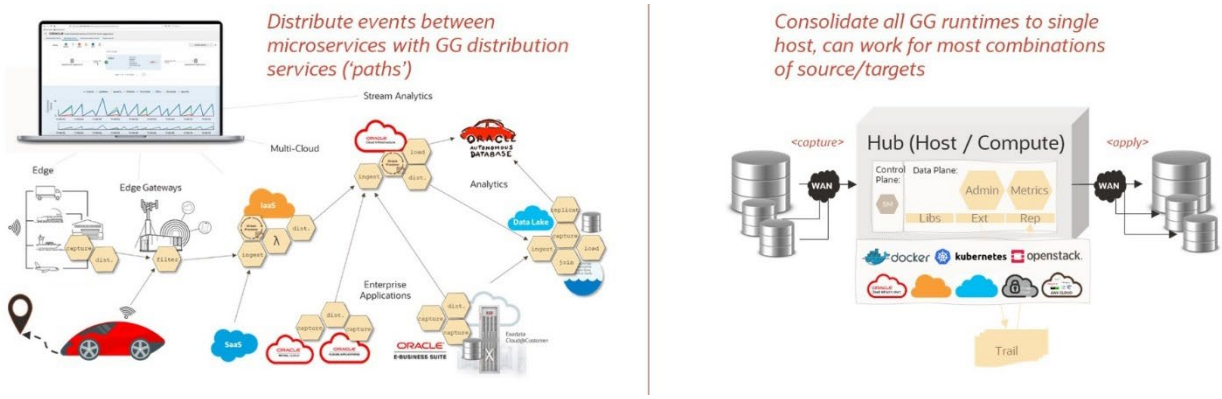

Figure 5: Oracle GoldenGate may be deployed as a Hub or in a microservices based Data Mesh

Oracle GoldenGate also can run as a mesh for larger enterprises, distributed environments or multi-cloud replication projects. With the Data Mesh approach, customers may distribute events between microservices layers/applications using GoldenGate distribution services.

General Advantages of Oracle GoldenGate:

- **Broadest database datatype and feature support -** Oracle GoldenGate is deeply integrated with the Oracle Database kernel (logging, redo, backup and recovery layer, security, administration, data guard, data vault, autonomous database etc.) and supports the widest set of Oracle Database datatypes.
- **Trust, Stability and Maturity -** Oracle GoldenGate is the most secure, stable, performant and reliable CDC / replication solution for Oracle. Oracle GoldenGate has integration with Oracle Clusterware for node resilience,

ACFS and DBFS storage for disaster recovery, and is certified with many 3rd party cloud HA/DR solutions (AWS EFS OneRegion, Google Cloud Hyperdisks, Azure Site Recovery) for resiliency in the cloud.

- **Highest performance -** Oracle GoldenGate uses private APIs that are capable of extremely low-impact capture, filtering, routing, transformation, and secure WebSocket's for very fast delivery of transactional data.
- **Oracle Maximum Availability Architecture -** Due to being the only CDC / replication tool certified for Oracle's Platinum Tier of maximum availability architecture, customers can be confident that GoldenGate is tested for the most stringent RPO (recovery point objective) and RTO (recovery time objective) scenarios – our customers trust GoldenGate with their most valuable and sensitive data.
- **Security and Data Protection –** GoldenGate provides end to end encryption of data at rest and in motion using AES 256 and FIPS-140 compliant algorithms. Communication to and from GoldenGate and between GoldenGate servers is conducted using TLS 1.3 and WSS encryption.
- **Modern Architecture -** GoldenGate is the only CDC / replication tool that keeps pace with new and emerging innovations (security, features, datatypes, etc) in the core Oracle Database.
- **Widest range of use cases** from conventional OLTP data replication and high availability (unidirectional, bi-directional, peer-to-peer and more) to data lake ingestion, or multi-cloud ingestion, SaaS application replication, messaging replication. During the last five years GoldenGate has been expanding into the stream processing patterns for use cases like data pipelines, data transformation, time-series analysis, predictive analysis, geospatial, real-time analytics etc.
- **Cloud Competency -** GoldenGate is the only CDC / replication technology certified for Autonomous Databases, Exadata, Exadata Cloud Service and Exadata Cloud at Customer. GoldenGate is also an integral part of the Oracle Database @ Azure and GCP partnerships to provide a fully managed OCI GoldenGate experience within 3rd party clouds.
- **Packaged Application Support -** Oracle GoldenGate works best with Oracle packaged applications like Cerner, Primavera, PeopleSoft, Siebel, and many SaaS applications (Fusion ERP, OTM, Retail, etc) running in OCI. It is the only logical replication product certified for Oracle e-Business Suite.
- **24x7 World-Wide Technical Support and development teams –** Oracle has development staff and technical support teams around the globe for 24x7 support.

From an operational point of view, GoldenGate has a command line and a graphical web user interface to support database administrators, system administrators, and DevOps teams. GoldenGate microservices benefits include:

- Simple to use web interface
- Fast and easy installation or provisioning
- Command line access via AdminClient that can run anywhere
- RESTful APIs for DevOps
- Offers built-in monitoring but also can be easily adapted to roll your own monitoring

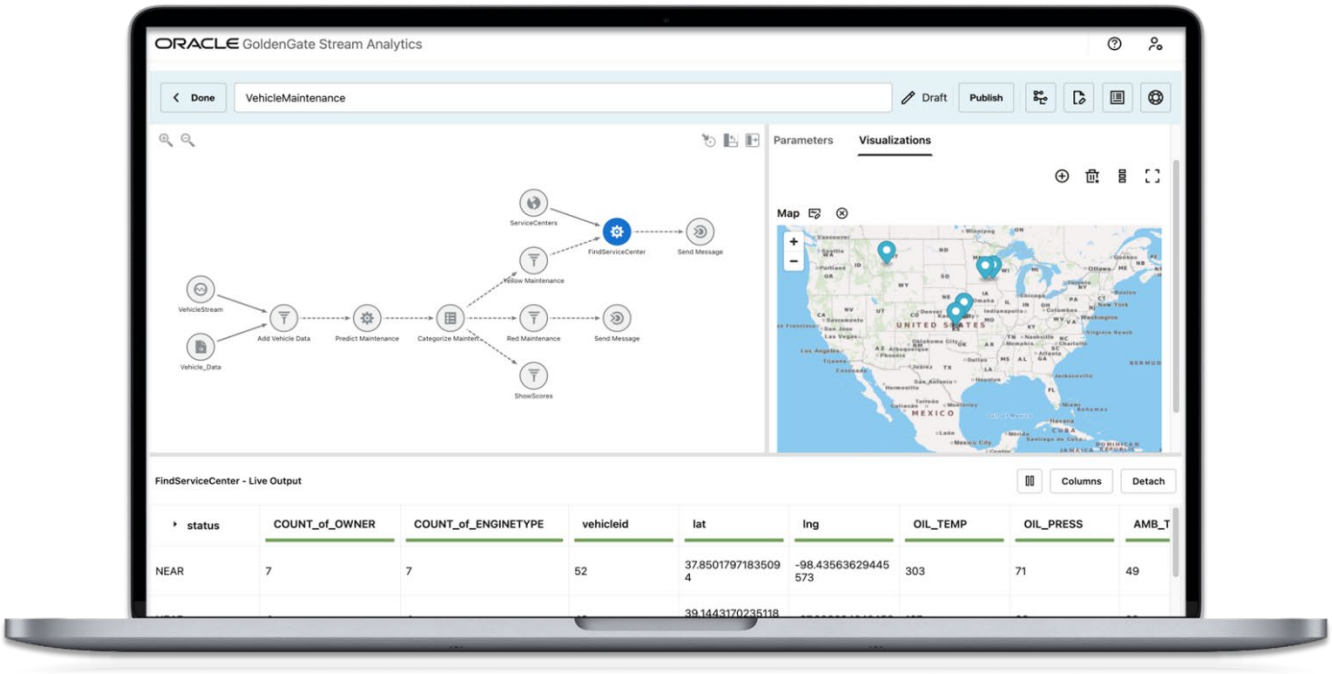

Figure 6: Simple and easy user interface for Oracle GoldenGate (both on-premises or cloud-native)

Banking and finance customers have always demanded the most rigorous security framework. GoldenGate security capabilities include:

- Data can be encrypted on disk and over the wire protocol
- Encryption profiles encapsulate the configuration information to retrieve a master key from Oracle Key Vault
- Native support for DMZ environments and private cloud environments.
- Supports Oracle Transparent Data Encryption (TDE) and Tablespace Encryption (TSE), through all versions of the database (third-party vendors with proprietary disk readers can't support this)
- Rich support for certificates across distributed services, including support for mTLS 1.3
- APIs are protected by Cross Site Request Forgery (CSRF) authentication. The default configuration is to enforce CSRF-token based protection.

Changed Data using XStream API

XStream is a strategic part of the Oracle Database, providing an SDK that enables 3rd party vendors to create products that capture from or deliver to the Oracle database. As LogMiner functionality changed to better suit Oracle Technical Support, the XStream API was made available as an SDK to provide high speed, log based capture and streamlined delivery for the Oracle Database.

XStream SDK consists of two major features: *XStream Out* and *XStream In*. XStream Out provides Oracle Database components and APIs that enable you to share data changes made to an Oracle database with other systems. XStream Out can retrieve both data manipulation language (DML) and data definition language (DDL) changes from the redo log and send these changes to a client application that uses the API. (Documentation)

In general, the XStream APIs are a very powerful, modern and up-to-date access layer for high-speed transactions replicating in or out of the Oracle Database. However, there are some important topics to be aware of.

Important XStream Facts

- **Licensing** - The use of XStream APIs require Oracle GoldenGate for Oracle licenses for any database where XStream is used, see here for more information: https://apex.oracle.com/database-features/
- **Database Queueing** - XStream will internally buffer uncommitted operations and only send data events once a commit has been processed for that transaction. This can impact the memory usage and overall performance of the database, especially in situations where the API is processing large transactions.
- **Developmental Overhead –** While the SDK is able push out or receive data events, it is up to the client application to build in transactional consistency, proper checkpointing for recovery, encryption of data at rest and in motion, and complex data types like user defined data types and LOB data (XML, CLOB, BLOB, etc).
- **Consistency of third-party tools** is not guaranteed by XStream. There is no way for the XStream APIs to guarantee high quality CDC when third-party tools are merely clients to the XStream API. Implementations may vary from one vendor to the next, and open-source frameworks may have altogether different implementations on top of XStream. Be alert to any vendor consistency issues with their use of XStream, even if Oracle Database customers implement a variety of CDC tools which all use XStream, the individual capabilities, quality and results will differ between the tools.
- **Tight integration with the Oracle Database** means that bug fixes, enhancements, and features may require certain versions of the database. Running the XStream API on databases running packaged applications (Oracle e-Business Suite, Cerner, SAP) that are limited to certain database versions may limit functionality or prevent the end user from applying bug fixes.
- **Technical Support** on XStream used by third-party tools is done by the third-party directly. If it is a bug in their code, they can provide the fix directly to the end user. If it's in the XStream API, they will open an SR with Oracle Support to request a bug fix from the database team. These bug fixes are triaged and handled according to Oracle Software Technical Support Policy (link). If a bug fix is provided, the end user will need to apply the database patch to resolve the issue.

Changed Data using Oracle Database LogMiner

Historically Oracle created LogMiner and maintained it for support and diagnostic use cases. Although many third-party vendors have adopted this diagnostic tool for change data capture, that use case was not the original intent of the API. Oracle redo logs record changes to database files so that database recovery operations can be performed and to conform with relational database ACID (Atomicity, Consistency, Integrity, Durability) properties.

LogMiner is an embedded diagnostic utility within the Oracle database that Oracle Support teams use to view the contents of redo log files to help audit and diagnose database issues related to log data.

Among the key capabilities of LogMiner are:

- Tracks when a logical corruption to a database, such as errors made at the application level, may have begun. It is important to know exactly when an error was made so that you know when to initiate time-based or change-based recovery. (documentation)
- Perform fine-grained recovery using a set of "SQL_UNDO" statements to reverse one or more DML operations.
- Database tuning using its trend analysis capabilities. Determine which table received the most updates and inserts to have a historical perspective on disk access statistics.
- Check transactional consistency because it can track not only the DDL and DML but also the order of the execution and which user executed the statements.

LogMiner is a powerful utility for the Oracle Database, but it also has some challenges to be aware of when attempting to stretch the use cases for LogMiner beyond what it was originally designed for.

Challenges with LogMiner

The following section lists some potential pain-points that any customer should be aware of when using LogMiner directly, or any third-party CDC tool based on LogMiner:

- LogMiner was designed as a diagnostic tool for the database redo logs. It will perform sub-optimally when used to mine the redo logs in an ongoing fashion.
- LogMiner is single-threaded and not designed for low overhead, high volume CDC.
- LogMiner will stop when it reached the current System Commit Number (SCN). New requests will start from the beginning of the logs similarly to a full-table-scan (FTS) operation. (documentation)
- It may be cumbersome or error prone when dealing with rollbacks or partial rollbacks when using LogMiner for CDC. When an application session is aborted (for instance, it exits suddenly), a rollback should occur for any transaction failure. If a transaction is cancelled or fails, then the database must cleanup the uncommitted work that was done by this transaction so that other transactions can progress. This cleanup involves rolling back the uncommitted work. From LogMiner point of view, the rollback statement is reported by itself as SQL_REDO and not SQL_UNDO. For a SQL which rolls back, no undo SQL is generated, and the rollback flag is set.
- LogMiner functionality in Autonomous Database has been limited to what is needed by Oracle Support for troubleshooting issues related to redo logs or data. Full details can be found here.
- LogMiner has a limited mount of datatype support and is incompatible with many of Oracle's new features. The supported data type list excludes most new data types introduced in Oracle 19c, and all new data types in Oracle 23ai. It lacks support for many commonly used database features like long column/table names, JSON Duality Views, and AI Vector data types (documentation)

A important fact is that each vendor implements their own LogMiner-based client has their own distinct limitations and liabilities – the use of LogMiner does not guarantee any particular level of uniformity or common expectations for service levels (SLAs) and KPIs.

For example, a third-party vendor who launched CDC support with LogMiner described the following limitations:

- Tables larger than 100 GB can't be backfilled.
- Oracle multi-tenant architecture (CDB/PDB) is not supported.
- Oracle Autonomous Database is not supported.
- Events from tables that do not have a primary key won't contain the information required to perform an update on the consumer side.
- Events have a size limitation of 3 MB.
- Index-organized tables (IOTs) are not supported.
- For columns of type BFILE, only the path to the file will be replicated. The contents of the file will not be replicated.
- Columns of data types ANYDATA, BLOB, CLOB, LONG/LONG RAW, NCLOB, UDT, UROWID, XMLTYPE are not supported, and will be replaced with NULL values.
- Oracle Label Security (OLS) is not replicated.
- If a schema changes, some events from the new schema may be read while the old schema is still applied.
- Not all changes to the source schema can be detected automatically, in which case data corruption may occur. The following schema changes may cause data corruption or failure to process the events downstream:
    - Dropping columns
    - Adding columns to the middle of a table
    - Changing the data type of a column
    - Reordering columns
    - Dropping tables (relevant if the same table is then recreated with new data added)
    - Truncating tables
- Materialized views created while the stream is running aren't backfilled automatically.

Log Miner remains focused on diagnostic use cases, and customers who choose third-party CDC tools that depend on LogMiner should proceed with caution. Use cases for online workloads that exceed hundreds of transactions per second may result in latency, memory and scalability issues. For small and low volume databases, LogMiner might be "good enough" for simple workloads. However, in many cases LogMiner will be paired with additional third-party tooling that can incur additional lifecycle challenges or unnecessary load on the source database.

Anyone using third-party CDC tools with LogMiner should carefully measure the impact and load requirements placed on the source database itself, and make sure that the datatype coverage, functional limitations and security concerns are ok for the business. As other diagnostic tools for the Oracle database mature, it may become unnecessary for Oracle to maintain LogMiner and functionality may be reduced or limited at any time as we did with the Oracle Autonomous Database.

Changed Data using Database Triggers

Using triggers can be an acceptable way to detect DML events for a small number of tables, and for use cases that do not have a lot of transaction volume. However, each trigger places new objects in the database and is essentially running a stored procedure for each event that is triggered – thus, there can be quite a bit of overhead on the application and the database when using triggers extensively. Few high-end tools will use triggers, but nonetheless some mainstream ETL tools still provide developer tools that setup and configure triggers as part of an ETL flow, this is something to beware of as it could negatively affect performance of the database.

This example creates a DML trigger that uses conditional predicates to determine which of its four possible triggering statements fired it:

```
CREATE OR REPLACE TRIGGER t
  BEFORE
    INSERT OR
    UPDATE OF salary, department_id OR
    DELETE
  ON employees
BEGIN
  CASE
    WHEN INSERTING THEN
      DBMS_OUTPUT.PUT_LINE('Inserting');
    WHEN UPDATING('salary') THEN
      DBMS_OUTPUT.PUT_LINE('Updating salary');
    WHEN UPDATING('department_id') THEN
      DBMS_OUTPUT.PUT_LINE('Updating department ID');
    WHEN DELETING THEN
      DBMS_OUTPUT.PUT_LINE('Deleting');
  END CASE;
END; /
```

Figure 7: example DML trigger

Brief Note Regarding Third-Party Disk Log Readers

Over the years some vendors have chosen to create proprietary software that reverse engineers Oracle Database logs directly to find data changes. Vendors who have chosen this strategy have avoided supported APIs and utilities, such as LogMiner and XStream. This puts your data, data security, and ongoing support at risk.

These proprietary log parsers have these shortcomings:

1. **Unsupported and Undocumented** – Oracle does not document the content and structure of the redo logs / archive logs and the use of 3rd party tools that reverse engineer these logs is unsupported and in violation of the Oracle Master Agreement (pt.8) and the Oracle Software License Agreement (pt.9)
2. **Lack of Lifecycle Integration** – Oracle can change the structure of the redo logs at any time, in any patch, without notification. It is up to the 3rd party vendor to ensure that their code is aware of how each Oracle patch may or may not make changes that introduce limitations, or completely prevent their log reader from working.
3. **Significant Functional Limitations** – since these readers are reverse engineering disk logs, they only see a fraction of the semantics (the meaning) of what the database engine is performing. This leads to many incompatibilities with Exadata, data compression (EHCC), table encryption, access to security keys, datatype incompatibilities etc. Vendors with binary log readers will force customers to disable this functionality in the database in order to process the data which can lead to lower database performance and increased security vulnerability.
4. **Governance and Security Risk for End Users** – administrative tools that require direct-host access to database logs and have the power to use user supplied encryption keys pose a significant compliance risk for several policies such as SOC2, Basel, BCBS239 and others.
5. **Risk for Unplanned Outage** – since this is an unsupported integration, Oracle may change the log formats, encryption, key vaults, etc at any time, causing unplanned outages and complex rollbacks when Database patches are applied.
6. **Problems with Cloud Databases** – Oracle managed cloud databases (e.g. Oracle Database Cloud Service, Oracle Autonomous Database, Oracle Exadata Cloud Service, and Oracle Exadata Cloud at Customer) may not be supported. In particular, it is not supported to install 3rd party binary log parsers from Oracle-managed cloud hosts. Further, Exadata cloud systems require encrypted tables, which are not readable by binary log parsers in Database 19.1 and higher.
7. **Mediocre Performance and Scale** – compared to GoldenGate's native access to Oracle database engine, any direct disk-based access to the redo logs causes unnecessary I/O issues and limitations on parallelism and compute utilization. Real-world workloads may run 2x-5x faster and handle 3x-10x more data volume when using native integrated GoldenGate extracts. These problems are further exacerbated when ASM storage is remotely accessed.
8. **Ownership Rights and Restrictions** – the Oracle Master Agreement expressly forbids causing or permitting others to reverse engineer data structures of the software, see section 3.4 of the Online Transactional Oracle Master Agreement, here: https://www.oracle.com/a/ocom/docs/lic-online-toma-us-eng-v091120.pdf
9. **Database Software License** – the Oracle Software License Agreement also expressly forbids causing or permitting others to reverse engineer data structures of the software, see section D, 3rd bullet, here: https://www.oracle.com/us/corporate/contracts/olsa-services/olsa-renewals-en-us-v053012-1867431.pdf

Summary – Oracle GoldenGate is the best CDC for Oracle Database

It should not be a surprise that the Oracle engineering teams are best equipped to also provide the best overall CDC and replication solution for the Oracle Database. GoldenGate is recognized by customers and analysts as the most functionally complete, highest performing, most trusted, and reliable data integration solution for the Oracle Database, as well as for the 100's of additional supported data platforms.

In this document we have explained numerous limitations and considerations for third-party CDC technologies working with Oracle databases. This table summarizes those CDC approaches.

[best] ●◉◉⊙○ [worst]	GoldenGate	XStream	LogMiner	Triggers	Disk Readers
Developer Experience					
Low Code Development	●	○	○	○	3rd party
Microservices APIs & Mesh Deployments	●	○	○	○	limited
Built-in Monitoring and Metrics	●	○	○	○	3rd party
Elastic, Multicloud Service	●	○	○	○	limited
Best for Oracle Databases					
Highest Performance (capture and replication)	●	◉	⊙	○	◉
Remote Capture/Apply	●	●	●	●	◉
Datatype Support	●	●	⊙	◉	⊙
Database HA / Recovery Capabilities	●	◉	◉	⊙	⊙
Certified for Oracle Max Availability	●	○	○	○	○
Multitenant, Container & Pluggable DB Support	●	◉	⊙	◉	⊙
Oracle Compression Support	●	●	●	●	○
Security (cloud DBs, TDE, wallets, vaults etc)	●	●	●	●	⊙
Heterogeneous Data Ecosystem					
Non-Oracle Databases	●	○	○	○	varies
Mainframes (DB2/z, Tandem, etc)	●	○	○	○	varies
Big Data, NoSQL, Kafka etc.	●	○	○	○	varies
Other					
Supported (Oracle will take SR on issues)	yes	yes	yes	yes	no
Licensing	GoldenGate	GoldenGate	Oracle DB	Oracle DB	unlicensed

Figure 8: Comparison of Oracle GoldenGate to alternative change capture APIs and approaches

In summary, GoldenGate surpasses all other third-party alternatives. We hope that this analysis helps you choose the right replication solutions for change data capture streams.

Additional Resources

Oracle GoldenGate Documentation:

- Oracle GoldenGate directory for Oracle Family
- Oracle MAA and GoldenGate Best Practices

Oracle GoldenGate Blogs

Oracle GoldenGate LiveLabs

Oracle GoldenGate YouTube Channel

Connect with us

Call +**1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

blogs.oracle.com facebook.com/oracle twitter.com/oracle

Authors: Oracle Product Development