ORACLE

Oracle GoldenGate Advantages

Choosing the Right CDC Solution for Oracle Databases

Executive Summary

Not all Oracle Database CDC is created equal. The method used to read change data from Oracle determines source impact, throughput, transaction ordering guarantees, datatype coverage, recovery semantics, security model, cloud compatibility, and long-term supportability. Choosing a CDC method is an architectural decision, not just a connector choice.

This paper describes three strategic approaches for reading committed change data from Oracle databases — Oracle LogMiner APIs, Oracle XStream APIs, and Oracle GoldenGate Native APIs — and explains how they differ in capability, performance, and fit. It also covers unsupported binary log readers as a separate risk category that warrants careful evaluation before any strategic deployment.

CENTRAL THESIS

The question is not simply whether a tool can read Oracle changes. The question is how it reads them, what guarantees it preserves, and what happens under failure, at scale, and when Oracle Database is patched or upgraded. For Oracle AI Database 26ai, GoldenGate 26ai Native APIs are the strongest option for high-volume, mission-critical, transactionally consistent CDC.

The three strategic options form a hierarchy:

- **LogMiner APIs** — a public Oracle Database utility, originally designed for diagnostics and recovery analysis. Accessible without additional licensing, which makes it attractive to third-party CDC tools. Appropriate for low-volume, simple, or exploratory use cases. Not the recommended architecture for high-volume, continuous, mission-critical replication.
- **XStream APIs** — a strategic Oracle Database API for change data streaming. Stronger performance and ordering guarantees than LogMiner. Licensed through Oracle GoldenGate. Used by qualified partners including Snowflake, Confluent, and IBM. A strong option when consumed through a well-implemented partner solution, with the understanding that XStream is an API layer — not a complete replication platform.
- **GoldenGate Native APIs** — private Oracle-internal APIs co-developed with the Oracle Database kernel. Not available to any third-party tool. The deepest integration path for Oracle Database CDC, delivering the highest throughput, lowest source impact, and strongest operational guarantees. Aligned with Oracle AI Database 26ai across Exadata, RAC, ASM, Autonomous Database, multitenant/PDB capture, and MAA Platinum and Diamond tier certification.

Oracle GoldenGate 26ai also supports downstream mining: organizations running older Oracle Database versions can forward redo logs to a newer Oracle AI Database 26ai instance where GoldenGate performs mining. This allows the replication infrastructure to operate on current database releases independently of the application lifecycle.

A note on scope of this paper: not every workload or use case requires the highest tier of transactional semantics/fidelity. Exploratory analytics, trend reporting, and many aggregate use cases tolerate latency or eventual consistency without material risk. The challenge arises when architectures designed for approximate workloads are reused for operational decisions, financial reporting, AI-driven workflows, or regulatory systems that require committed-state accuracy.

This paper is about understanding which approach is appropriate for which workload — and knowing where the gaps become material.

Table of Contents

Oracle CDC Is Not a Commodity Decision 4
Evaluation Criteria for Oracle CDC 5
Summary Decision Path 6
Approach 1 — Oracle LogMiner APIs 8
Approach 2 — Oracle XStream APIs 9
Approach 3 — Oracle GoldenGate Native APIs 11
Risk Category — Unsupported Binary Log Readers 15
Side-by-Side Comparison 16
When GoldenGate May Be More Than You Need 17
Practical Buyer Guidance 18
Conclusion 20

Appendices:

Appendix A: Sample PoC Checklist 21
Appendix B: FAQ about CDC with Oracle Database 26
Appendix C: Glossary of Terms 31
Appendix D: Citations and References 39

Oracle CDC Is Not a Commodity Decision

Oracle Database maintains a redo log for its own recovery purposes. Every committed transaction — inserts, updates, deletes, schema changes — is written to that log in commit order. Reading from that log is the foundation of all log-based CDC for Oracle [1]. But how you read it — through which API, with what level of database integration, with what guarantees about ordering, completeness, and recovery — determines almost everything about the quality of the downstream data.

What Oracle Transaction Logs Record

Oracle redo logs capture every committed database event: DML (insert, update, delete), DDL (schema changes), procedure executions, and the metadata needed to reconstruct the exact state of the database at any point. The System Commit Number (SCN) provides a globally ordered sequence across all concurrent transactions, which is the foundation for any system that claims to preserve commit order.

When data moves from Oracle into a downstream system — a lakehouse, a streaming platform, an AI vector database, a secondary Oracle instance — the question is what properties of the original committed transactions survived the journey. Three properties are most consequential:

- **Commit order:** Did changes arrive downstream in the same sequence they committed at the source?
- **Transaction boundary integrity:** Did all row changes belonging to a single source transaction arrive together — or did they scatter across batches and micro-batches, creating inconsistent intermediate states downstream?
- **Recovery semantics:** After a failure, does the pipeline restart from an exact position without duplicating already-delivered transactions or silently dropping events?

Why These Properties Matter for AI and Analytics

For traditional BI reporting, occasional staleness or a minor ordering inconsistency produced a wrong chart that a human could question. For AI systems that execute workflows autonomously — approving credit, detecting fraud, generating regulatory filings, personalizing customer interactions — the same data quality failures produce wrong actions at machine speed.

An AI fraud model that receives transaction events out of commit order will evaluate behavioral patterns against a temporal mosaic, not the actual sequence of customer actions. A RAG-based customer service agent grounded in yesterday's batch extract answers questions about an account state that may no longer exist. A regulatory reporting system built on a pipeline that cannot prove source-to-target fidelity cannot satisfy audit requirements that increasingly require demonstrable, not asserted, data accuracy.

These are not edge cases. They are the predictable failure modes of CDC architectures built for approximate workloads and reused for mission-critical ones.

KEY INSIGHT

Choosing a CDC method is an architectural decision, not just a connector choice. The method determines source impact, scalability, transaction ordering, recovery behavior, security model, and long-term compatibility with new Oracle Database features. Evaluating only whether a tool 'can read Oracle changes' misses the more consequential questions about how it reads them and what guarantees it preserves.

Evaluation Criteria for Oracle CDC

The table below provides a practical evaluation framework for comparing Oracle CDC approaches. Not every criterion carries equal weight for every use case — the right weighting depends on whether the workload is mission-critical or exploratory, high-volume or low-volume, cloud-managed or on-premises. The framework is intended to surface the questions a buyer should ask, not to declare a single winner for all scenarios.

EVALUATION CRITERION	LOGMINER	XSTREAM	GOLDENGATE NATIVE	BINARY LOG READERS
Oracle supported?	Yes	Yes	Yes	No
Strategic / roadmap-aligned?	Limited	Yes	Yes	No
Commit-order guarantee?	Not guaranteed	Yes	Yes	Not guaranteed
Transaction boundary integrity?	Varies by tool	Depends on implementation	Yes	Not guaranteed
Throughput / scalability	Low-medium	Medium-high	Highest	Varies
Source system impact	Medium-high	Low-medium	Lowest	Varies
RAC / ASM / Exadata support	Limited	Yes	Yes	Problematic
Autonomous Database support	Yes	No	Yes	No
Multitenant / PDB capture	Limited	Yes	Yes (per-PDB)	No
TDE / encryption support	Yes	Yes	Yes	Problematic

Licensing	Included in DB license	Requires GoldenGate or partner license	Oracle GoldenGate	Unsupported; license risk
Best-fit workloads	Diagnostics; low-volume; simple CDC	Partner integrations; strategic ecosystem	Mission-critical; high-volume; MAA	Avoid for strategic CDC

Note: assessments above are general characterizations. Specific implementations vary. The XStream and GoldenGate Native API rows describe Oracle's supported behavior. Third-party tools built on LogMiner or XStream introduce additional variability depending on implementation quality.

Summary Decision Path

Use the lowest-risk CDC architecture that satisfies the actual workload. The right choice depends less on whether a tool can read Oracle changes and more on what the workload requires the CDC pipeline to preserve under scale, failure, patching, and operational change.

Start With Five Questions:

1. **How critical is the downstream use case?**
   If the data supports financial settlement, regulatory reporting, fraud detection, operational decisioning, customer-facing AI, disaster recovery, or other systems where wrong data produces material risk, prioritize transaction correctness, recoverability, supportability, and demonstrable fidelity over connector convenience.

2. **How demanding is the workload?**
   If the source database produces high redo volume, large transactions, bursty batch loads, frequent schema changes, or low-latency requirements, evaluate CDC approaches against sustained throughput, p95/p99 lag, source impact, and recovery behavior under realistic production load.

3. **Which Oracle features are in use?**
   RAC, ASM, Exadata, TDE, Data Guard, Autonomous Database, Multitenant/PDBs, advanced datatypes, JSON/OSON, VECTOR, JSON Relational Duality Views, blockchain tables, and immutable tables all raise the bar. A CDC tool should be evaluated against the Oracle environment actually in production, not against a simplified test schema.

4. **What target semantics are required?**
   Moving committed Oracle transactions into Kafka, a lakehouse, object storage, a warehouse, a vector database, or another Oracle database creates different ordering, idempotency, replay, and schema-evolution requirements. The CDC method must be evaluated end-to-end, including the target behavior.

5. **Who supports the pipeline when something breaks?**
   A strategic CDC architecture should have clear support boundaries across Oracle Database, the CDC platform, the target platform, and any cloud service involved. If a database patch, encryption change, failover event, or datatype expansion breaks capture, the support path should be explicit before production deployment.

Practical Rules to Follow:

- Choose LogMiner-based CDC when the use case is diagnostic, exploratory, low-volume, or non-critical, and when datatype coverage, source impact, restart behavior, and schema handling have been tested and accepted for that specific tool.

- Choose XStream-based CDC when a qualified partner solution uses XStream correctly, the target platform is part of that partner architecture, licensing is clear, and the partner implementation has demonstrated checkpointing, monitoring, recovery, schema handling, and security at the level the workload requires.

- Choose Oracle GoldenGate Native APIs when the workload is high-volume, mission-critical, regulated, AI-driven, cloud-modernization oriented, or dependent on advanced Oracle Database features. GoldenGate should be the primary architecture to evaluate when correctness, low source impact, MAA alignment, Oracle feature coverage, and long-term roadmap compatibility are material.

- Treat unsupported binary log readers as a risk category, not as a strategic Oracle CDC approach. They should be considered only after the technical, operational, security, licensing, and legal implications of unsupported redo parsing have been explicitly reviewed.

Approach 1 — Oracle LogMiner APIs

What LogMiner Is

Oracle LogMiner is a component of Oracle Database that enables users to query online and archived redo log files through a SQL interface [2], [3]. It was originally designed as an Oracle support and diagnostic tool — useful for tracking when a logical corruption occurred, performing fine-grained recovery, analyzing historical database activity, and understanding transaction ordering for auditing purposes.

LogMiner reconstructs SQL_REDO statements from the redo logs, representing what was committed. This makes it readable by applications and third-party tools through standard Oracle database connections, without requiring additional Oracle software licensing — which is a significant reason for its popularity with third-party CDC vendors.

Why Third-Party CDC Tools Use It

LogMiner's accessibility is its primary advantage for third-party CDC vendors. Because it exposes redo log content through a standard SQL interface over JDBC/ODBC, any tool with a database connection can implement Oracle CDC without Oracle-specific integration work. This lowers the barrier to Oracle support for a vendor's product considerably.

It is important to understand, however, that each vendor who builds on LogMiner implements their own client above it. There is no standard SLA, no guaranteed datatype coverage, and no uniform behavior across vendors for edge cases like long transactions, rollbacks, DDL changes, or schema evolution. Choosing a LogMiner-based CDC tool means evaluating that specific vendor's implementation, not LogMiner itself.

Technical Concerns for Continuous CDC

LogMiner was not designed to be a continuous, high-volume CDC engine. Several of its characteristics create challenges when used for this purpose:

- **Single-threaded design:** LogMiner is not architected for parallelism. For high-transaction-rate databases, it becomes a bottleneck. Third-party tools working around this limitation may impose additional overhead on the source database through repeated mining sessions or log scanning.
- **Continuous_mine desupported in Oracle 19c:** The continuous_mine option, which allowed LogMiner to read from redo logs as they were generated, was deprecated in Oracle Database 12.2 and desupported in Oracle Database 19c [2]. Any third-party tool that relied on this mode requires a different implementation for customers running Oracle 19c and later — which covers the vast majority of current Oracle deployments.

- **Log restart behavior:** LogMiner stops when it reaches the current System Commit Number (SCN). New requests must restart from the beginning of the available logs, similar to a full-table-scan operation. This creates latency and resource pressure for continuous workloads.
- **Rollback handling:** When a transaction is rolled back, LogMiner reports the rollback as SQL_REDO without generating corresponding SQL_UNDO. Tools that do not handle this correctly may propagate uncommitted state or misinterpret aborted operations.
- **SQL_REDO reconstruction is not native replay:** LogMiner reconstructs SQL statements from redo records. This is a transformation — the reconstructed SQL is not identical in semantics to the original committed operation, particularly for complex types. It is not the same as native access to the committed transaction stream.

Datatype and Feature Gaps

Beginning with Oracle Database 12.2, LogMiner generates 'unsupported' SQL_REDO and SQL_UNDO for a number of datatypes including: JSON, OSON, BFILE, nested tables, objects with nested tables, tables with identity columns, temporal validity columns, PKREF columns, long identifiers, PKOID columns, and nested table attributes [2]. Third-party tools vary widely in how they handle or work around these gaps.

Third party tools built on LogMiner also typically do not support Oracle Multitenant/PDB architectures fully, may not support Oracle Autonomous Database, and face compatibility challenges with Exadata-specific compression (EHCC), Transparent Data Encryption (TDE), and ASM storage.

Best Fit and Not Best Fit

LogMiner is appropriate when the use case is diagnostic, investigative, or genuinely low-volume: recovering a deleted record, auditing historical activity, understanding what transactions occurred during an incident window, or evaluating CDC for a development or test environment.

LogMiner-based CDC is not the recommended architecture for enterprise-grade, continuous CDC from high-volume Oracle database systems — particularly where latency, datatype completeness, schema evolution, RAC/Exadata/Autonomous compatibility, or recovery semantics are material requirements.

Approach 2 — Oracle XStream APIs

What XStream Is

Oracle XStream is a strategic Oracle Database API layer that enables client applications to receive data changes from an Oracle database (XStream Out) and send data changes into an Oracle database

(XStream In). XStream Out reads committed changes from Oracle redo logs and exposes them as Logical Change Records (LCRs) — structured representations of committed DML and DDL operations — to a connected client application [10].

XStream shares underlying technology with Oracle GoldenGate and is specifically designed for ISVs, partners, and advanced technical users who need a supported Oracle API for high-speed transaction streaming. Unlike LogMiner, XStream provides guaranteed commit ordering and is designed for continuous, high-performance CDC workloads. It is a meaningful step up from LogMiner in both performance and operational semantics.

XStream In Practice: What It Provides and What It Does Not

XStream is an API layer. It provides a supported, high-performance path to Oracle's redo stream and delivers LCRs to a connected client. What it does not provide is the application layer above it: checkpointing and recovery logic, lifecycle management, schema evolution handling, error handling and alerting, monitoring, operational automation, security, and management of restart behavior after failures.

All of that is the responsibility of the tool or implementation built on top of XStream. Two products that both claim 'XStream-based Oracle CDC' may differ substantially in quality, feature coverage, and failure behavior depending on how well each has implemented those higher-level capabilities. Evaluating an XStream-based product means evaluating the complete implementation, not just the API it uses.

Partner Ecosystem: Validated Public Implementations

XStream has a growing partner ecosystem. The following implementations are publicly documented and represent the kind of qualified partner deployments where XStream is the right choice:

- **Snowflake Openflow Connector for Oracle:** Generally available as of February 2026. Reads Logical Change Records directly from the Oracle XStream outbound server queue in near real time and streams changes into Snowflake via Snowpipe Streaming. Agentless, zero-footprint architecture. Supports Oracle 12cR2 through 26ai on-premises, Exadata, OCI, and AWS RDS Custom for Oracle. Available with an embedded Snowflake-managed XStream license or BYOL with an existing Oracle GoldenGate license [11], [12].
- **Confluent Cloud Oracle XStream CDC Source Connector:** Fully managed connector for Confluent Cloud. Captures committed changes from Oracle Database using the XStream API and represents them as change event records in Apache Kafka topics. Supports Oracle Database 12c and later, with recent updates adding Amazon RDS for Oracle support [13], [15].
- **IBM Data Replication (IIDR) for Oracle:** IBM IIDR has a standing partnership with Oracle around XStream Out and uses it as the foundation for high-performance Oracle CDC in IBM's data replication stack [14].

XStream Licensing

XStream generally requires an Oracle GoldenGate license for any database where it is used [4]. Some approved commercial partners may include or resell XStream access as part of their own commercial arrangement with Oracle — the Snowflake Openflow Connector is an example of a partner offering an embedded XStream license to customers directly. Organizations should verify their specific licensing obligations with Oracle before deploying any XStream-based tool.

Best Fit and Not Best Fit

- XStream is the right middle tier in the Oracle CDC hierarchy: strategic, supported, and partner-friendly. It is a strong option when:
- A qualified partner solution (Snowflake, Confluent, IBM, or similar) uses XStream correctly.
- The use case is to stream Oracle CDC into that partner's target platform.
- The partner has implemented checkpointing, lifecycle management, schema handling, error recovery, security, and operational monitoring correctly.
- XStream licensing and lifecycle responsibilities are clearly understood by the customer.

XStream is not designed to provide the same operational management depth, topology flexibility, native trail architecture, data verification, integrated monitoring, MAA patterns, or performance ceiling as GoldenGate Native APIs. For the most demanding Oracle Database CDC workloads — high volume, low latency, mission-critical, multi-target, MAA-certified — GoldenGate Native APIs remain the reference architecture.

Approach 3 — Oracle GoldenGate Native APIs

What Makes Native APIs Different

Oracle GoldenGate 26ai uses private, Oracle-internal APIs for change data capture that are not available through any public database interface — not through LogMiner, not through XStream, and not through any third-party tool. These APIs integrate directly with the Oracle Database kernel, including the SGA (System Global Area) layer, and have been co-developed within Oracle Database across every major release since 11.2.0.4.

This is not a matter of degree. Oracle exposes private, high-speed kernel APIs to GoldenGate that are not available through any public interface (including XStream or LogMiner). The SGA internals and GoldenGate integrated capture APIs represent more than fifteen years of joint development between the GoldenGate and Oracle Database engineering teams — a depth of integration that no competitor

can access and that no third-party tool can replicate. Ultimately, none of these tools are inside the kernel. GoldenGate is.

KEY INSIGHT

GoldenGate 26ai is not just the best-supported Oracle CDC tool. It is the only CDC platform engineered inside the Oracle Database product boundary — built with the same team, aligned with the same roadmap, and tested under the same mission-critical conditions. For Oracle AI Database 26ai, that is the architecture that matters.

What GoldenGate Integrated Extract Does

GoldenGate Integrated Extract reads committed transactions directly from Oracle databases using the private GoldenGate Native API — not by querying application tables, not through SQL reconstruction, and not by reading disk-resident log files with a binary parser [5]–[9]. The Extract process reads change records in the order they were committed and transmits them, with full transactional context, through GoldenGate's native binary trail protocol.

Because the API reads directly from Oracle's internal transaction buffers, GoldenGate imposes the smallest possible load on the source application. No SQL queries hit live application tables. No batch jobs compete for I/O. No table locks are required. This non-invasive characteristic is not a marketing claim — it is an architectural consequence of reading from a stream the database was already writing for its own recovery purposes.

Transaction Guarantees

- **Commit-order preservation:** GoldenGate delivers changes downstream in the exact sequence they committed at the source. The SCN-ordered commit stream from the integrated capture API provides a globally consistent ordering baseline that no LogMiner-based or XStream-based tool can exceed.
- **Transaction boundary integrity:** All row changes from a single source transaction arrive at the downstream target together as a group. A payment that updates an account balance, a ledger entry, and a risk position in a single atomic operation arrives downstream as a coherent business event — not scattered across batches.
- **Exactly-once delivery where supported and configured:** GoldenGate's checkpoint architecture maintains a precise position in both the source transaction log and the target apply queue. After any failure — network partition, process crash, host reboot, rolling database upgrade — GoldenGate restarts from the exact checkpoint without duplicating already-delivered transactions or dropping any events.
- **Complete DML and DDL capture:** GoldenGate captures DDL schema changes alongside DML. Downstream targets stay structurally synchronized with source systems without

manual schema migration work — a capability that is critical as Oracle AI Database 26ai introduces new datatypes and table structures [17].

Oracle AI Database 26ai Alignment

GoldenGate 26ai captures and replicates Oracle-specific data constructs that no third-party tool can handle correctly — precisely because they require kernel-level awareness of Oracle's internal structures. These include:

- **Per-PDB capture in Oracle Multitenant:** GoldenGate scopes change data capture to individual Pluggable Databases (PDBs) rather than mining at the Container Database (CDB) level. This preserves tenant isolation boundaries and simplifies compliance in multi-tenant deployments. These are meaningful security benefits for large enterprises.
- **Oracle Autonomous Database:** GoldenGate is the only replication platform currently supported for high performance CDC from Oracle Autonomous Database, where system-level access is restricted and alternative approaches are not viable [19].
- **Oracle AI Vector Search — VECTOR datatype replication:** GoldenGate captures and replicates the Oracle VECTOR datatype and supports inline embedding generation during replication via integration with OCI Generative AI, enabling real-time vector pipeline architectures directly from committed Oracle transactions [16].
- **JSON Relational Duality Views:** GoldenGate captures changes from JSON Relational Duality Views directly from the transaction log and can publish fully materialized document representations to downstream consumers, not just row-level deltas [18].
- **Blockchain and immutable tables:** Starting with Oracle AI Database 26ai, GoldenGate supports replication of blockchain and immutable tables, including the associated metadata required for auditability. Correct replication requires awareness of their specialized semantics — which a kernel-integrated tool can provide reliably [16].

Performance and Scale

GoldenGate's kernel-level capture path and parallel apply architecture produce performance profiles that are structurally unreachable for tools operating through JDBC connections or third-party log parsing. A few significant examples include:

- **Tier-one global bank:** Captures more than 4 TB of Oracle Database transaction logs per hour in a continuous GoldenGate pipeline — with consistently low impact on the application's system performance.
- **Oracle lab benchmark:** GoldenGate 26ai on a quarter-rack Exadata X11M demonstrates more than 2.5 million rows/second of changed data processed with Parallel Replicat [28]
- **Typical end-to-end latency:** Source commit to target apply is measured in milliseconds to single-digit seconds in well-tuned deployments, not minutes or hours. Network latency is typically the gating factor, not GoldenGate software.

MAA Certification, Exadata, RAC, ASM, and Data Guard

GoldenGate 26ai is certified for Oracle Maximum Availability Architecture (MAA) at the Platinum and Diamond tiers — the highest levels of Oracle's availability framework. These tiers require deep integration with Oracle's internal recovery, consistency, and failover mechanisms. GoldenGate is the only CDC platform certified at these tiers [20].

GoldenGate also provides native integration with Oracle Real Application Clusters (RAC), Automatic Storage Management (ASM), Data Guard, Exadata (including Exadata Cloud Service, Exadata Multicloud and Exadata Cloud@Customer), and Oracle Autonomous Database [24]. These environments present specific challenges — particularly around Exadata encryption (EHCC), ASM storage access, pluggable database isolation and redo log management across RAC nodes — that GoldenGate's kernel integration handles natively and that third-party tools frequently cannot address fully.

Security Model

GoldenGate's security architecture was designed for financial services environments from the beginning. It supports AES-256 encryption for data at rest (trail files) [22] and leverages post-quantum–resilient cryptographic mechanisms, including ML-KEM within TLS 1.3, for data in motion [23]. It supports Oracle Transparent Data Encryption (TDE) and tablespace encryption through all versions of the database — a capability that creates significant challenges for tools that read logs from disk without kernel-level access to the Oracle key vault [21].

Best Fit and Not Best Fit

GoldenGate 26ai Native APIs are the right choice for:

- High-volume, continuous CDC from Oracle databases.
- Mission-critical replication requiring MAA Platinum or Diamond certification.
- Oracle Exadata, RAC, ASM, Autonomous Database, and multitenant/PDB workloads.
- Downstream AI pipelines, vector databases, and RAG architectures requiring current, ordered, committed Oracle data.
- Regulatory reporting, financial settlement, and audit-grade data movement where source-to-target fidelity must be demonstrable.
- Multi-target architectures where a single capture process feeds multiple downstream consumers simultaneously.

Risk Category — Unsupported Binary Log Readers

Some third-party vendors bypass Oracle-supported APIs entirely and attempt to read Oracle redo and archive log files directly from disk storage, reverse engineering Oracle's internal log formats without Oracle's cooperation or support [26]. This approach is architecturally distinct from LogMiner, XStream, and GoldenGate Native APIs and represents a different category of risk entirely.

How Binary Log Readers Work

These tools mount direct access to Oracle's on-disk redo logs or archive logs and parse the binary format without using any Oracle-supported interface. Because Oracle does not document the internal structure of redo log files, these parsers reverse engineer what they can observe — which is a subset of the full semantic content of the database's transaction stream.

Technical and Operational Risks

- **Unsupported access path:** Oracle does not support this approach. When a database patch, upgrade, encryption change, or internal format update breaks the parser, Oracle has no obligation to provide assistance or a fix. The vendor is entirely on its own.
- **Undocumented format dependency:** Oracle may change redo log formats, encryption, compression, key vault structure, or internal storage semantics at any time — at any patch level, not just major releases. A binary log parser that works today may fail after a routine quarterly patch.
- **Incompatibility with Exadata and ASM:** Exadata smart scan, EHCC compression, and ASM storage present specific challenges for disk-level log readers. Vendors typically require customers to disable these features to use their product, eliminating the performance and storage benefits of Exadata infrastructure.
- **TDE and encryption limitations:** Tools that read redo logs directly from disk require access to Oracle encryption keys to decrypt the log content. This creates a security architecture in which individuals operating the CDC tool have access to master encryption keys — a pattern that conflicts with SOC2, BCBS 239, and financial services security frameworks. Exadata Cloud systems in particular require encrypted tables that binary log parsers in Database 19c and higher cannot access.
- **Cloud database incompatibility:** Oracle-managed cloud databases — Oracle Autonomous Database, Oracle Exadata Cloud Service, Oracle Exadata Cloud at Customer — may not permit installation of third-party binary log parsers. The architecture of managed cloud services explicitly restricts the kind of host-level access these tools require.
- **No MAA certification:** There is no Oracle Maximum Availability Architecture configuration that certifies binary log readers for failover, restart, or disaster recovery automation. Mission-critical deployments have no certified path for lifecycle management.

- **Performance and parallelism limits:** Disk-based log reading creates unnecessary I/O overhead and limits the parallelism available for log processing. Real-world workloads using GoldenGate's integrated capture may process 2x-5x more data volume with lower source impact, particularly when ASM storage is accessed.

Legal and Licensing Considerations

The Oracle Master Agreement expressly addresses reverse engineering of data structures [27]. Organizations considering binary log reader tools should obtain independent legal review of their specific contract terms before deployment. This paper does not constitute legal advice, and the specific applicability of Oracle license terms to a given tool or deployment should be evaluated by qualified counsel with access to the actual agreements in question.

Summary: Treat as a Risk Category

For pre-sales evaluation purposes, binary log readers should be categorized as a risk rather than a supported Oracle CDC strategy. The absence of Oracle support, the dependency on undocumented internal formats, the security and regulatory implications of key exposure, and the incompatibility with managed cloud and Exadata environments collectively make this approach unsuitable for strategic Oracle CDC deployments.

Side-by-Side Comparison

EVALUATION CRITERION	LOGMINER	XSTREAM	GOLDENGATE NATIVE	BINARY LOG READERS
Oracle supported?	Yes	Yes	Yes	No
Strategic / roadmap-aligned?	Limited	Yes	Yes	No
Commit-order guarantee?	Not guaranteed	Yes	Yes	Not guaranteed
Transaction boundary integrity?	Varies by tool	Depends on implementation	Yes	Not guaranteed
Throughput / scalability	Low-medium	Medium-high	Highest	Varies
Source system impact	Medium-high	Low-medium	Lowest	Varies
RAC / ASM / Exadata support	Limited	Yes	Yes	Problematic

Autonomous Database support	Yes	No	Yes	No
Multitenant / PDB capture	Limited	Yes	Yes (per-PDB)	No
TDE / encryption support	Yes	Yes	Yes	Problematic
MAA Certified	No	No	Yes	No
Fully-managed Cloud	Depends	Depends	Yes	Problematic
AI/Vector Support	Depends	Depends	Yes	Problematic
Licensing	Included in DB license	Requires GoldenGate or partner license	Oracle GoldenGate	Unsupported; license risk
Recommendation	Diagnostics; low-volume; simple CDC	Partner integrations; strategic ecosystem	Mission-critical; high-volume; MAA	Avoid for strategic CDC

When GoldenGate May Be More Than You Need

Not every Oracle CDC use case requires the deepest integration path. A diligent CDC evaluation should distinguish between workloads that require mission-critical transaction fidelity and workloads where an alternative tool or target-specific approach is sufficient.

GoldenGate may be more than the workload requires when the source database is small, the transaction rate is modest, the data is used for exploratory analytics, and the business can tolerate latency, limited datatype coverage, or occasional manual intervention. A development pipeline that moves changes from a small Oracle schema into a test warehouse does not have the same risk profile as a production system feeding fraud detection, financial reconciliation, regulatory reporting, or real-time AI workflows.

GoldenGate may also be more than necessary when the organization is already standardizing on a qualified partner platform that provides an Oracle XStream-based connector for a narrow, well-defined target use case. For example, if the only requirement is to stream Oracle changes into a specific managed cloud analytics or streaming platform, and that partner has implemented XStream licensing, checkpointing, restart behavior, monitoring, schema handling, and support escalation correctly, the partner solution may be appropriate.

Similarly, some LogMiner-based tools can be acceptable for low-volume or non-critical use cases, especially where the schema is simple, Oracle feature usage is limited, and the team has tested the

tool's behavior against the specific database version, datatypes, DDL patterns, supplemental logging requirements, and recovery scenarios involved. In these cases, the key is not whether LogMiner can support every enterprise CDC requirement. It is whether the specific implementation satisfies the limited workload in question.

The danger is not choosing a lighter-weight approach for a lighter-weight workload. The danger is allowing a tool selected for convenience, price, or a narrow analytics use case to become the default CDC architecture for systems that require committed-state accuracy, low latency, source-to-target fidelity, auditability, and production-grade recovery.

A useful decision rule is this: if the downstream system can tolerate approximate, delayed, or manually recoverable data, then GoldenGate may be optional. If the downstream system must be trusted as an extension of the committed Oracle source, GoldenGate should be evaluated first.

Practical Buyer Guidance

Choose LogMiner APIs if...

- The use case is diagnostic: recovering a deleted record, auditing historical activity, or investigating what happened during an incident window.
- The workload is genuinely low-volume, non-critical, or exploratory — such as a development environment, a proof-of-concept, or a small database where the limitations of LogMiner-based tools are acceptable.
- The datatype coverage and schema evolution behavior of the specific tool have been evaluated against the target database's specific use of Oracle features.
- The operational impact on the source database has been measured and is acceptable for the workload's performance requirements.

Choose XStream APIs if...

- You are deploying an approved partner solution (Snowflake Openflow, Confluent Platform, IBM IIDR, or similar) that has implemented XStream correctly with full checkpointing, lifecycle management, and error handling.
- The use case is to stream Oracle CDC into that partner's target platform and the partner's implementation quality has been evaluated — not just the fact that XStream is used.
- XStream licensing and lifecycle responsibilities are clearly understood by both the vendor and the customer, and the appropriate commercial arrangements are in place.
- The partner implementation covers the specific Oracle Database features in use — including RAC, multitenant/PDB, and TDE if relevant.

Choose GoldenGate 26ai (native) if...

- The workload is mission-critical: financial settlement, regulatory reporting, high availability/disaster recovery, or any use case where downtime or data loss has direct financial or regulatory consequences.
- High throughput, low latency, or transactionally consistent CDC from Oracle AI Database 26ai is required — particularly at volumes where LogMiner and XStream performance ceilings become material.
- Oracle Exadata, RAC, ASM, Autonomous Database, or multitenant/PDB environments are involved.
- The downstream consumer is an AI pipeline, vector database, or agentic workflow that requires current, ordered, committed Oracle data to function correctly.
- MAA Platinum or Diamond tier certification is required for the data movement layer.
- Source-to-target data fidelity must be demonstrable through algorithmic verification (GoldenGate Veridata) rather than assumed [25].
- Low-cost options such as GoldenGate Free or Oracle Cloud Infrastructure (OCI) GoldenGate are available to your organization.

Treat Binary Log Readers with Caution

- The tool requires you to mount storage for remote parsing.
- The tool requires direct host-level access to Oracle redo or archive logs.
- The tool requires you to share Master Encryption Keys across users
- The tool requires you to disable Exadata features, encryption, or ASM to function.
- The use case involves Autonomous Database, Oracle-managed cloud services, or Exadata Cloud configurations.

ASK YOUR CDC VENDOR THESE TEN QUESTIONS

1. Which Oracle API do you use to read changes: LogMiner, XStream, or something else?
2. If you use XStream, how do you manage XStream licensing with your customers?
3. If you use LogMiner, how do you handle the continuous_mine desupport in Oracle 19c?
4. How do you preserve transaction boundaries when replicating to non-ACID targets?
5. How do you handle rollbacks and partial rollbacks?
6. How do you handle DDL schema changes — column additions, drops, renames, type changes?
7. How do you recover after a network failure or pipeline restart without duplicating or skipping transactions?

8. Do you support Oracle RAC, ASM, Exadata, Autonomous Database, TDE, and multitenant/PDB capture?
9. Can you prove that source and target data are identical? What verification mechanism do you use?
10. Who provides support when Oracle Database is patched or upgraded and your CDC pipeline breaks?

Conclusion

Oracle Database CDC is not a trivial concern to be overlooked by tech leadership. The method used to read change data from Oracle databases determines everything downstream: latency, throughput, transactional correctness, security model, cloud compatibility, operational lifecycle, alignment with the database's evolving feature set and ultimately your ability to trust the data in your AI and reporting systems.

The three strategic approaches form a clear hierarchy for mission-critical workloads. LogMiner is useful for the diagnostic and low-volume purposes it was designed for. XStream is strategic, supported, and partner-ready — and a strong option when consumed through a well-implemented partner solution that addresses all the capabilities XStream itself does not provide. GoldenGate 26ai Native APIs are the deepest integration path available for Oracle Database CDC — the only approach engineered inside the Oracle Database product boundary, co-developed with Oracle database engineering across every release, certified for MAA Platinum and Diamond tiers, and aligned with the AI-era capabilities that define the next generation of Oracle database workloads.

For organizations evaluating CDC vendors, the safest question is not 'Can this tool read Oracle changes?' Almost anything can read Oracle changes. The question is: 'Which Oracle-supported path does it use, and what guarantees does that path preserve under load, failure, and change?'

GoldenGate 26ai answers that question with thirty years of production history in the world's most demanding Oracle environments — and with a product roadmap aligned to Oracle AI Database 26ai in a way that no third-party tool can replicate.

> ***"For Oracle AI Database 26ai, GoldenGate 26ai is the reference architecture for high-volume, mission-critical, transactionally consistent CDC. That is not a marketing claim — it is an architectural fact grounded in kernel-level integration that no alternative can access."***

Appendix A: Sample PoC Checklist

Use this checklist to evaluate whether a CDC architecture is suitable for high-volume, mission-critical Oracle Database workloads. The goal is not simply to confirm that changes can be captured, but to prove that the solution preserves correctness, performance, recoverability, and operational trust under real-world conditions.

This checklist is intentionally vendor-neutral. It tests properties that any Oracle CDC solution should be able to demonstrate; it does not assume a particular tool. Items shown in bold are disqualifying for mission-critical workloads: a failure on any one of them means the architecture is not production-ready, regardless of how well it performed elsewhere.

Step 0 — Define the Test Before You Run It

The most common way a POC misleads is by running on a quiet, non-representative system and discovering the ceiling only in production. Establish the targets and environment first.

- ☐ Capture peak and sustained redo generation rates from production (e.g., from AWR / Statspack), not from a quiet dev system. *Record:* _____
- ☐ Inventory the Oracle features actually in use in the target environment: RAC, ASM, Exadata, Multitenant/PDB, Autonomous, Data Guard, TDE, and the specific datatypes in production.
- ☐ Define the pass thresholds before testing: acceptable replication lag (avg / p95 / p99), acceptable source-system impact, and required recovery behavior.
- ☐ Size the POC environment to within a realistic factor of production volume, or document the extrapolation method and its limits.

1. Workload and Throughput Validation

- ☐ Confirm peak redo generation rate during normal and high-volume periods. *Record:* _____
- ☐ Test sustained CDC capture at expected production transaction volume. *Record:* _____
- ☐ Test burst volume above expected peak (quarter-end, batch load, AI feature-refresh windows). *Record:* _____
- ☐ Measure source database CPU, memory, I/O, and network impact during capture, against a no-CDC baseline. *Record:* _____
- ☐ Measure replication lag under load — average, p95, and p99 — not just average. *Record:* _____
- ☐ Confirm whether latency remains acceptable during long-running and high-row-count transactions. *Record:* _____

2. Transaction Correctness

- ☐ **Validate commit-order preservation across concurrent transactions.**
- ☐ **Confirm transaction boundary integrity — partial transactions must not appear downstream.**
- ☐ Test large transactions with many row changes.
- ☐ **Test rollback and partial-rollback behavior; confirm rolled-back changes are never delivered as committed.**
- ☐ Test mixed insert, update, delete, and merge workloads.
- ☐ **Confirm exactly-once or effectively-once delivery behavior where required and supported.**
- ☐ Validate restart behavior after CDC process failure, database restart, and network interruption.

3. Schema and DDL Handling

- ☐ Test table creation, alteration, and deletion scenarios.
- ☐ Test column add, drop, rename, and datatype-change scenarios.
- ☐ Test primary-key and index changes.
- ☐ Confirm behavior when DDL occurs during active DML workloads.
- ☐ Confirm DDL replication propagates automatically, without manual intervention on the target.
- ☐ Confirm supplemental logging requirements and measure their overhead on the source. *Record:* _____
- ☐ Confirm how unsupported or partial DDL is reported — paused, skipped, or flagged for remediation — never silently dropped.

4. Oracle Database Feature Coverage

- ☐ Test CDC against Oracle Multitenant / PDB environments, including per-PDB scoping where required.
- ☐ Validate capture from RAC databases under normal operation.
- ☐ Test RAC node failure and service relocation.
- ☐ Validate ASM and Oracle-managed storage compatibility.
- ☐ Test Data Guard switchover / failover behavior if applicable.
- ☐ Validate Exadata-specific configurations (including EHCC compression) if the source runs on Exadata.
- ☐ Confirm compatibility with Autonomous Database if applicable.
- ☐ Test TDE-protected tablespaces and encrypted databases.

- ☐ **Confirm coverage for every Oracle datatype and structure used in production — including JSON / OSON, XML, LOBs, object types, spatial, VECTOR, JSON Relational Duality views, and blockchain / immutable tables where present.**

5. Recovery and Resilience

- ☐ Stop the CDC process mid-workload and restart it.
- ☐ Simulate a target-system outage and confirm catch-up behavior.
- ☐ Simulate network interruption between source, CDC service, and target.
- ☐ Validate checkpointing and recovery from the last confirmed position.
- ☐ **Confirm no duplicate, missing, or reordered transactions after any recovery scenario.**
- ☐ Test archive-log availability requirements during extended outages. *Record:* _____
- ☐ Confirm alerting when logs are missing, inaccessible, or purged before they are read.

6. Initial Load and Cutover to CDC

- ☐ Validate the initial load / instantiation of the target from the source.
- ☐ **Confirm a zero-gap, zero-overlap handoff from initial snapshot to ongoing change capture at the boundary SCN.**
- ☐ Measure time-to-complete for initial load at production data size. *Record:* _____
- ☐ Confirm the source remains usable, with acceptable impact, during initial load. *Record:* _____

7. Security and Compliance

- ☐ Confirm a least-privilege access model for CDC users and service accounts.
- ☐ Validate encryption in transit.
- ☐ Validate encryption at rest for captured data, trail files, staging areas, and queues.
- ☐ **Determine whether the solution requires CDC operators to access Oracle Master Encryption keys (MEK), and whether that satisfies your separation-of-duties, SOC 2, and BCBS 239 requirements.**
- ☐ Review whether the solution requires direct host-level access to redo logs, archive logs, or storage volumes.
- ☐ Confirm handling of sensitive data, regulated fields, and masking requirements.
- ☐ Confirm auditability of CDC configuration changes, operator actions, and access events.

8. Operational Manageability

- ☐ Confirm monitoring for lag, errors, throughput, checkpoint position, and process health.
- ☐ Validate alerting integration with existing operations tooling.
- ☐ Confirm visibility into failed transactions, skipped records, and retry behavior.
- ☐ Test controlled pause and resume.
- ☐ Test planned-maintenance procedures.
- ☐ Confirm upgrade procedures for the CDC platform itself.
- ☐ Validate how configuration changes are promoted across dev, test, and production.
- ☐ Confirm the support model, escalation path, and vendor responsibility boundaries — including who is accountable when an Oracle patch or upgrade breaks the pipeline.

9. Target and Use-Case Validation

- ☐ Test delivery to each required target: data warehouse, lakehouse, Kafka, object storage, application database, or AI pipeline.
- ☐ Validate target-side ordering and idempotency.
- ☐ Confirm data-format compatibility for downstream consumers.
- ☐ Test multi-target fan-out if required.
- ☐ Validate transformation, filtering, and routing behavior.
- ☐ Confirm downstream consumers receive enough metadata for lineage, governance, and replay.

10. AI, Analytics, and Governance Readiness

- ☐ Confirm CDC latency supports the AI, analytics, or operational-decisioning requirement. *Record:* _____
- ☐ Validate that feature stores, vector pipelines, and RAG systems receive committed, consistent data.
- ☐ Confirm lineage metadata is available for captured changes.
- ☐ Validate that reconciliation / data-quality processes can detect missing or divergent records.
- ☐ Confirm how the solution supports governance controls for sensitive or regulated AI inputs.
- ☐ **Confirm that stale, partial, or out-of-order changes cannot reach AI or reporting systems.**

11. Final Evaluation

- ☐ Did the solution preserve transaction correctness under realistic load?
- ☐ Did it meet latency requirements without unacceptable source impact?
- ☐ Did it handle the Oracle features actually used in the real environment?
- ☐ Did it recover cleanly from failures without data loss or duplication?
- ☐ Did it provide enough monitoring, security, and auditability for production?
- ☐ Is the solution supported by Oracle, the CDC vendor, and the target-platform vendor?
- ☐ Is the architecture appropriate for the criticality of the workload?

Verdict rule: if any bolded item above (correctness, recovery, security, or coverage of an Oracle feature in production use) failed, the architecture is not production-ready for a mission-critical workload — no matter how strong its throughput or latency numbers. For mission-critical systems, the decisive POC result is not that capture works once, but that it stays correct, recoverable, observable, and supportable under stress.

Appendix B: FAQ about CDC with Oracle Database

This appendix answers the questions buyers and architects ask most often when evaluating change data capture for Oracle Database. The answers follow the same principle as the body of this paper: the issue is not whether a tool can read Oracle changes, but how it reads them and what guarantees survive under load, under failure, and when Oracle Database is patched or upgraded.

1. What is CDC in Oracle Database?
Change data capture (CDC) is the process of identifying inserts, updates, deletes, and — depending on the approach — DDL changes made in an Oracle Database and making those changes available to downstream systems. In Oracle environments, log-based CDC reads from the redo stream, because redo already records every committed change in commit order for the database's own recovery purposes.

For enterprise workloads, CDC is not merely a data-movement technique. As the body of this paper argues, it is an architectural decision that determines transaction correctness, source impact, latency, recoverability, security, and long-term supportability.

2. Why is Oracle CDC different from CDC on other databases?
CDC is different on every database, there is no standard for how databases should handle CDC. Oracle Database has a rich internal architecture: redo, standby redo, LCRs, XIDs, SCNs, RAC, ASM, TDE, Exadata, Multitenant/PDBs, Autonomous Database, advanced datatypes, and Oracle-specific recovery semantics. A connector that handles simple row changes on a commodity database may be inadequate for high-volume Oracle systems that depend on commit ordering, rollback handling, DDL capture, encrypted storage, or per-tenant isolation.

For this reason, Oracle CDC should be evaluated against the actual Oracle topology and workload in question — not against the generic claim that a tool "can read Oracle changes."

3. What are the main Oracle CDC approaches?
There are three supported strategic approaches, plus one risk category:

- LogMiner APIs — a public Oracle Database utility that exposes redo content through a SQL interface.
- XStream APIs — a strategic Oracle Database API that streams Logical Change Records to a client application.
- GoldenGate Native APIs — private, Oracle-internal capture APIs co-developed with the database kernel.
- Unsupported binary log readers — tools that parse Oracle redo or archive logs directly, outside any supported Oracle API. This is a risk category, not a strategic approach.

The right choice depends on workload volume, database version, required correctness, Oracle feature usage, operational risk, and support expectations.

4. Is LogMiner a CDC product?
No. LogMiner is an Oracle Database utility for querying online and archived redo through a SQL interface; it was designed for diagnostics and recovery analysis, not as a continuous, high-volume replication platform. It can underpin CDC, but it requires supplemental logging before redo can be analyzed usefully, and Oracle Database does not enable that logging by default.

LogMiner is well suited to diagnostic, exploratory, or genuinely low-volume use. It should be evaluated carefully before being relied on for mission-critical, high-throughput, or regulated workloads.

5. What happened to LogMiner continuous_mine?
The continuous_mine option for DBMS_LOGMNR.START_LOGMNR was deprecated in Oracle Database 12.2 and desupported in Oracle Database 19c, which covers the vast majority of current deployments. Some older LogMiner-based CDC designs depended on continuous mining, so any LogMiner-based evaluation should verify explicitly how the tool now handles log switching, archive-log availability, restart, latency, and long-running capture without it.

6. What is supplemental logging, and why does it matter?
Supplemental logging adds information to redo so downstream systems can identify and reconstruct changed rows correctly. Oracle Database does not provide it by default, and at least minimal supplemental logging must be enabled for redo to be analyzed.

For CDC, the logging level is a design trade-off. Too little can make downstream reconstruction unreliable; too much increases redo volume and source overhead, and identification-key logging can introduce cursor invalidation. A proof of concept should measure both correctness and performance impact under realistic load.

7. Does LogMiner support every Oracle datatype and table structure?
No. Beginning with Oracle Database 12.2, LogMiner generates “unsupported” SQL_REDO and SQL_UNDO for a range of datatypes and structures — including JSON, OSON, BFILE, nested tables, identity columns, temporal validity columns, and certain object-related structures — and a single unsupported column can cause an entire table to be skipped.

This makes datatype and schema coverage one of the most important questions in any CDC evaluation: does the chosen architecture support the datatypes and schema patterns actually used in production?

8. What is XStream, and how is it different from LogMiner?
XStream is an Oracle Database API framework that lets client applications receive committed changes from Oracle. XStream Out reads committed changes from redo, converts them into Logical Change Records (LCRs), and delivers them through an outbound server to a connected client. Unlike LogMiner, XStream provides guaranteed commit ordering and is designed for continuous, high-performance streaming.

XStream is a meaningful step up from LogMiner, but it remains an API layer. The client or partner product must still implement checkpointing, restart, delivery semantics, monitoring, error handling, schema evolution, and target integration.

9. Is XStream the same thing as GoldenGate?
No. XStream is an Oracle Database API; GoldenGate is a complete replication and CDC platform. Several qualified partners — including Snowflake (Openflow), Confluent, and IBM (IIDR) — build Oracle CDC on XStream Out. Because the operational behavior lives in the partner implementation rather than in the API, an XStream-based product should be evaluated as a complete implementation, not merely on the fact that it uses a supported Oracle API.

10. Why is GoldenGate described as the reference architecture for Oracle CDC?
GoldenGate is Oracle's strategic replication platform, and its integrated capture uses private, Oracle-internal APIs co-developed with the database kernel across every major release. That integration is why GoldenGate handles RAC, ASM, TDE, Exadata, and Multitenant/PDBs more directly than generic approaches, and why it is the only CDC platform certified for Oracle Maximum Availability Architecture (MAA) at the Platinum and Diamond tiers.

For low-volume or non-critical use, GoldenGate may be more than the workload requires. For high-volume, mission-critical, regulated, AI, analytics, or cloud-modernization workloads, it is the reference architecture this paper recommends evaluating first.

11. Can Oracle CDC capture DDL changes?
It depends on the approach and configuration. XStream Out can retrieve both DML and DDL from redo, and GoldenGate captures DDL schema changes alongside DML so downstream targets stay structurally synchronized. The specifics depend on Oracle Database version, capture mode, target type, and configuration.

DDL should be tested explicitly in any proof of concept — table creation, column changes, index and partition changes, and schema evolution occurring during active DML.

12. Does CDC guarantee exactly-once delivery?
Not automatically. Redo preserves database transaction information, but end-to-end delivery semantics depend on the CDC architecture, checkpointing, target behavior, retry handling, and failure recovery. Some pipelines provide at-least-once delivery; some provide exactly-once or effectively-once behavior under specific configurations and target combinations.

The better question is not whether a tool claims exactly-once, but whether — after process failure, network interruption, target outage, and restart — it can demonstrate that no committed transaction was lost, duplicated, or applied out of order.

13. How does CDC handle rollbacks and long-running transactions?
A production CDC system must never publish rolled-back work as committed truth, and it must handle long-running transactions without excessive memory pressure, latency, or restart risk. Approaches differ here: LogMiner, for instance, reports rollbacks as SQL_REDO without

corresponding SQL_UNDO, which a tool must handle correctly to avoid propagating uncommitted state. Capture position, checkpoints, and redo availability all bear on recovery.

Long transactions and rollbacks should be an explicit part of any mission-critical CDC test plan.

14. What source-database impact should I expect from CDC?

Every CDC approach has some source impact; the question is how much and where. Supplemental logging increases redo volume, and identification-key logging can cause cursor invalidation. Capture can consume CPU, memory, I/O, or network resources, and tools that query application tables or repeatedly scan logs add further load. Approaches that read from the database's internal transaction stream — which Oracle is already writing for recovery — impose the least overhead.

A credible evaluation measures source CPU, memory, I/O, redo generation, archive-log pressure, and latency under production-like load.

15. What should I ask a CDC vendor before choosing a solution?

These questions expand on the vendor checklist in the body of this paper:

- Which Oracle CDC path does the product use — LogMiner, XStream, GoldenGate, or direct binary log access — and is that path supported by Oracle?
- How does it preserve commit order and transaction boundaries, including to non-ACID targets?
- How does it handle rollbacks, long transactions, and restart without loss or duplication?
- Which Oracle datatypes and table structures are unsupported?
- Does it support RAC, ASM, TDE, Exadata, Autonomous Database, and Multitenant/PDBs?
- How does it handle DDL and schema evolution?
- What delivery semantics are supported for my specific target?
- Can it prove source-to-target fidelity, and through what verification mechanism?
- Who provides support when the issue involves Oracle Database internals, or when a patch or upgrade breaks the pipeline?

16. Why does CDC matter for AI and analytics?

AI, analytics, and reporting increasingly act on fresh operational data. If the pipeline delivers stale, partial, duplicated, or out-of-order changes, downstream systems make decisions on incorrect state — a problem for fraud detection, customer-service agents, operational dashboards, financial reporting, feature stores, and retrieval-augmented generation. As the paper notes, at machine speed these data-quality failures become wrong actions, not just wrong charts.

This is also where Oracle AI Database 26ai alignment matters: GoldenGate captures and replicates the Oracle VECTOR datatype (with inline embedding generation via OCI Generative AI), JSON Relational Duality Views, and blockchain and immutable tables — constructs that require kernel-level awareness to replicate correctly. For AI workloads, the CDC decision is less about movement speed than about whether downstream intelligence can trust the data it receives.

17. What is the simplest rule of thumb?

Use the lowest-risk CDC architecture that satisfies the real workload. For simple, low-volume, or non-critical analysis, LogMiner-based CDC may be sufficient if its limitations are acceptable. For partner-managed platforms, XStream is appropriate when the partner's implementation has been validated. For high-volume, mission-critical, regulated, AI, analytics, or Oracle-modernization workloads, GoldenGate Native APIs should be the primary architecture to evaluate. Treat unsupported binary log readers as a risk category, to be considered only after the legal, operational, security, and support implications have been explicitly reviewed.

Appendix C: Glossary of Terms

This glossary defines terms used throughout the paper. Entries are scoped to their CDC and Oracle Database context rather than their broadest general definition. Where a term carries architectural significance — for example, where its absence or mishandling produces measurable data quality risk — that significance is noted.

ACID
The four transaction properties — atomicity, consistency, isolation, and durability — that define what it means for a database operation to commit correctly. Oracle Database provides full ACID semantics. A CDC pipeline that preserves ACID-committed transactions downstream delivers the same logical guarantees the application relied on at the source. A pipeline that does not — through partial delivery, out-of-order application, or dropped events — silently degrades the integrity of every downstream consumer.

Archive Log
A preserved copy of an Oracle redo log segment written after the active log filled and rotated. CDC tools that fall behind the current redo position must access archive logs to continue reading the transaction stream without gaps. The ability to mine archive logs correctly — including across log switches, outages, and recovery events — is a basic operational requirement for any continuous CDC pipeline.

ASM (Automatic Storage Management)
Oracle's integrated volume and file management layer for database files, including redo logs and archive logs. ASM abstracts physical storage behind Oracle-managed disk groups. CDC tools that require direct filesystem access to log files encounter ASM as an incompatibility barrier. GoldenGate's integrated capture accesses redo through Oracle's internal APIs and is fully compatible with ASM without requiring bypass or workaround.

Autonomous Database
Oracle's fully managed cloud database service. In Autonomous Database, system-level access is restricted by design — host access, agent installation, and direct log access are not permitted. CDC approaches that depend on any of those access paths are not viable in this environment. GoldenGate is currently the only CDC platform with a supported, high-performance capture path from Oracle Autonomous Database.

Binary Log Reader
A CDC tool that reads Oracle redo or archive log files directly from disk by reverse-engineering their internal binary format, without using any Oracle-supported API. Because Oracle does not document the redo log format, these parsers depend on reverse-engineered observations that may break at any patch level. In this paper, binary log reading is categorized as a risk rather than a supported CDC strategy. See: Risk Category — Unsupported Binary Log Readers.

CDC (Change Data Capture)
The process of identifying committed database changes — inserts, updates, deletes, and schema modifications — and making them available to downstream systems. The central argument of this paper is that CDC is not a commodity function: the method used to capture changes determines what guarantees survive the move from source to target, and those guarantees determine whether downstream systems can trust the data they receive.

CDB (Container Database)
The root container in Oracle Multitenant architecture, which hosts one or more Pluggable Databases. The CDB level is where Oracle manages redo logging for the entire multitenant instance. CDC tools that mine at the CDB level must handle correct tenant isolation to avoid mixing changes across PDB boundaries. See also: Multitenant, PDB.

Checkpoint
A durable record of the CDC pipeline's current position in the source transaction stream and, where applicable, the target apply queue. A well-designed checkpoint architecture allows a pipeline to restart from an exact position after any failure — without losing events, duplicating already-delivered transactions, or reordering committed changes. Checkpoint quality is one of the most consequential differences between CDC implementations that appear equivalent at a feature level.

Commit Order
The sequence in which transactions committed at the source database, as established by Oracle's System Commit Number. Preserving commit order downstream is required for any consumer that reconstructs database state, evaluates behavioral sequences, or performs time-sensitive analysis. AI models that process transaction sequences out of commit order evaluate patterns against a scrambled timeline — not the actual sequence of events.

Continuous Mining / continuous_mine
A historical LogMiner option that allowed redo to be read as it was generated, without restarting a mining session for each log segment. Deprecated in Oracle Database 12.2 and desupported in Oracle Database 19c. Any LogMiner-based CDC tool that relied on this mode has required a different implementation for customers running Oracle 19c and later — which covers the vast majority of current Oracle deployments. Tools that have not disclosed how they addressed this desupport warrant specific scrutiny.

Data Guard
Oracle's high availability and disaster recovery technology for maintaining synchronized standby databases. Data Guard switchover, failover, and role transitions introduce specific events — redo stream continuity, SCN synchronization, primary/standby role changes — that a CDC pipeline must handle correctly to avoid data loss or reordering at transition time. GoldenGate is certified for Data Guard environments and recovers correctly from switchover and failover events.

DDL (Data Definition Language)
SQL operations that modify database structure: CREATE TABLE, ALTER COLUMN, DROP INDEX, and similar statements. DDL changes are committed transactions that must be captured and applied downstream in the correct position relative to DML changes that precede or follow them. A CDC pipeline that applies DML correctly but ignores or mishandles DDL produces structural drift between source and target that compounds over time.

DML (Data Manipulation Language)
SQL operations that change data values: INSERT, UPDATE, DELETE, and MERGE. DML capture is the core function of CDC. The distinctions that matter in this paper are not whether a tool captures DML, but whether it captures the full set of committed DML operations correctly — including complex types, large objects, Exadata-compressed rows, and encrypted content — with correct transaction boundaries and commit ordering preserved.

Downstream Mining
A GoldenGate architecture in which redo log data is forwarded from the source Oracle database to a separate Oracle Database instance — typically a newer release — where GoldenGate performs the actual mining. This removes mining load from the production source and allows the replication infrastructure to operate on a current, fully supported database version independently of the application database's release lifecycle.

EHCC (Exadata Hybrid Columnar Compression)
Oracle's Exadata-specific compression technology that stores data in a hybrid row-columnar format to maximize compression ratios. EHCC operates at a level that is invisible to standard SQL but visible to kernel-level CDC. Binary log readers and other tools that parse redo without Oracle kernel integration encounter EHCC as an incompatibility that vendors typically address by requiring customers to disable the feature — eliminating a primary Exadata benefit.

Exactly-Once Delivery
A delivery guarantee in which each committed source transaction is applied to the target exactly once — not skipped and not duplicated — even across failures and restarts. Exactly-once behavior depends on the CDC product, configuration, checkpointing architecture, target system capabilities, and failure recovery design working correctly together. It is a property of the complete pipeline, not a checkbox feature of a single component.

Exadata
Oracle's engineered database platform for high-performance Oracle workloads, combining custom storage, networking, and compute with deep Oracle Database integration. Exadata is deployed on-premises and as Oracle Exadata Cloud Service, Exadata Cloud@Customer, and Exadata Multicloud. CDC compatibility with Exadata requires handling of ASM, EHCC compression, TDE, and Exadata-specific I/O patterns — capabilities that require kernel-level integration to address correctly.

Extract
The Oracle GoldenGate process responsible for capturing changes from the source database. In Oracle CDC contexts, Extract reads committed transaction changes through GoldenGate's integrated capture API and writes them to trail files for downstream processing. Integrated Extract is the preferred mode for Oracle-to-Oracle and Oracle-to-any-target pipelines — co-designed with Oracle Database engineering and compatible with RAC, ASM, TDE, multitenant/PDB, and Exadata environments.

GoldenGate Native APIs
Private Oracle-internal APIs used exclusively by Oracle GoldenGate for deep integration with Oracle Database internals, including the SGA layer. These APIs are not accessible through any public Oracle interface — not through LogMiner, not through XStream, and not through any third-party tool. They represent more than fifteen years of co-development between GoldenGate and Oracle Database engineering, and they are the foundation of GoldenGate's performance, ordering, and operational guarantee advantages over any alternative Oracle CDC approach.

GoldenGate Veridata
An Oracle GoldenGate product that compares source and target data algorithmically to verify that they are identical. Veridata provides demonstrable source-to-target fidelity rather than assumed correctness. In regulatory, financial, and audit-grade deployments, the distinction between asserted fidelity and verified fidelity is material — particularly when the data powers automated decisions or supports compliance reporting.

Integrated Extract
The GoldenGate capture mode that uses Oracle's integrated log mining server — a component embedded in Oracle Database and shared with GoldenGate through the GoldenGate Native API — to capture committed transaction changes. Integrated Extract is the recommended configuration for Oracle CDC and the only approach that delivers full compatibility with RAC, ASM, TDE, multitenant/PDB, and all Oracle AI Database 26ai features.

JSON Relational Duality View
An Oracle Database 23ai feature that exposes relational tables as JSON documents while maintaining relational storage and transactional consistency. Applications can read and write through the JSON interface; the underlying storage remains relational. CDC support for JSON Relational Duality Views is relevant for modern application architectures and AI-oriented pipelines that consume data in document form but require committed-state accuracy.

LCR (Logical Change Record)
The structured representation of a committed database change used by Oracle XStream and Oracle replication technologies. LCRs carry the row-level change data, the operation type, the commit SCN, and associated metadata for a committed DML or DDL operation. XStream delivers LCRs to connected client applications for downstream processing. The quality of what a tool does with an LCR — checkpointing, ordering, schema management, error recovery — determines the quality of the CDC pipeline above XStream.

LogMiner
An Oracle Database component that exposes redo log content through a SQL interface, originally designed for diagnostic and recovery analysis. LogMiner is accessible without additional Oracle licensing, which has made it the default foundation for third-party Oracle CDC tools. Its accessibility, however, does not make it equivalent to deeper integration approaches in throughput, ordering guarantees, datatype completeness, or operational resilience. See: Approach 1 — Oracle LogMiner APIs.

MAA (Maximum Availability Architecture)
Oracle's prescriptive framework for high availability, disaster recovery, and operational resilience across Oracle Database environments. MAA defines tiers — Bronze, Silver, Gold, Platinum, Diamond — each requiring progressively deeper integration with Oracle's internal recovery, consistency, and failover mechanisms. GoldenGate 26ai is the only CDC platform certified for MAA Platinum and Diamond tiers. Certification at these tiers is not self-declared — it requires demonstrated integration with Oracle's complete availability stack.

Multitenant
Oracle's database architecture in which a Container Database (CDB) hosts multiple Pluggable Databases (PDBs) that share Oracle infrastructure while maintaining logical isolation. Oracle Database 21c made multitenant the mandatory architecture. CDC solutions operating in multitenant environments must correctly handle tenant boundaries, per-PDB supplemental logging, and PDB-scoped capture to prevent cross-tenant data leakage and to deliver compliant, isolated replication streams.

OSON
Oracle's optimized binary JSON storage format, used internally to store JSON data efficiently in Oracle Database. OSON is not standard JSON text — it is an Oracle-specific binary encoding that tools without kernel-level access cannot decode correctly from redo. CDC tools that encounter OSON without proper support will either fail, skip, or misrepresent JSON column changes, degrading the fidelity of downstream data that depends on JSON content.

PDB (Pluggable Database)
A tenant database within an Oracle Multitenant Container Database. From an application perspective, a PDB presents as a self-contained Oracle database. From a CDC perspective, PDB-level capture allows change data extraction to be scoped to a specific tenant without mining changes from the entire CDB. GoldenGate supports per-PDB capture with correct tenant isolation — an architectural requirement for large enterprises running multiple business units or regulatory domains within a shared Oracle Multitenant infrastructure.

Parallel Replicat
A GoldenGate apply mode that processes and applies captured changes in parallel across multiple threads, while preserving transaction dependencies that require serialized application. Parallel Replicat is the mechanism behind GoldenGate's documented throughput benchmarks — including more than 2.5 million rows per second on a quarter-rack Exadata X11M — and is critical for high-volume replication workloads where apply throughput would otherwise constrain end-to-end pipeline latency.

RAC (Real Application Clusters)
Oracle's clustered database architecture in which multiple database instances share a single database, providing horizontal scalability and instance-level fault tolerance. RAC produces redo logs across multiple nodes that must be correlated correctly to reconstruct a globally ordered transaction stream. CDC tools that handle RAC incorrectly may produce out-of-order or incomplete change streams when instance failover or log switching occurs. GoldenGate Integrated Extract handles RAC redo correlation natively.

RAG (Retrieval-Augmented Generation)
An AI architecture in which a model retrieves relevant enterprise data at inference time before generating a response. The quality of RAG output is directly bounded by the currency and accuracy of the underlying data. A RAG system grounded in stale or inconsistently replicated data generates confident responses about a database state that may no longer exist. CDC that preserves committed-state accuracy keeps RAG knowledge stores current and transactionally correct.

Recovery Semantics
The defined behavior of a CDC pipeline when it resumes after failure. Strong recovery semantics means: the pipeline restarts from a known, durable checkpoint position; no committed events are lost; no already-delivered events are re-delivered; and the ordering of events is preserved across the restart boundary. Weak recovery semantics — at-least-once delivery without deduplication, or restart from an approximate position — produce duplication or gaps that downstream consumers must detect and handle, if they can detect them at all.

Redo Log
Oracle Database's primary transaction log, written for recovery purposes. Every committed operation — DML, DDL, procedure execution — is recorded in commit order. The redo log is the authoritative source of truth for what happened in the database and when. All log-based CDC for Oracle — whether through LogMiner, XStream, GoldenGate Native APIs, or binary log readers — ultimately reads from, or reads APIs derived from, this log.

Replicat
The Oracle GoldenGate process that reads captured changes from trail files and applies them to a target system. Replicat handles transaction reconstruction, apply ordering, conflict detection, and error management at the target. The Parallel Replicat mode is used for high-throughput workloads; standard Replicat is available for targets that require fully serialized apply.

Rollback
The cancellation of a transaction before it commits. Rolled-back work is not committed and must not appear in the downstream change stream. CDC tools must correctly distinguish between committed and rolled-back operations in the redo log. LogMiner reports rollbacks as SQL_REDO entries without generating corresponding SQL_UNDO, creating a specific handling challenge for tools built on it. Mishandled rollbacks propagate uncommitted state downstream — producing data that never existed in the source database.

SCN (System Commit Number)
A monotonically increasing logical timestamp assigned by Oracle Database to every committed transaction. SCNs establish a globally consistent ordering of all transactions across the database, including across RAC nodes. The SCN sequence is the foundation for commit-order preservation in CDC, for precise restart positioning after failures, and for point-in-time recovery operations. Any CDC approach that cannot reliably track and resume from SCN position cannot guarantee correct ordering or lossless recovery.

SGA (System Global Area)
Oracle Database's shared memory area used by all Oracle database processes. GoldenGate's integrated capture accesses the SGA directly through GoldenGate Native APIs, reading committed transactions from Oracle's in-memory structures rather than from disk-resident log files. This SGA-level integration is a key source of GoldenGate's throughput and source-impact advantages and is not accessible through any public Oracle API.

Source-to-Target Fidelity
The degree to which a downstream target accurately reflects the committed state of the source database. High fidelity means changes are complete, ordered, transactionally intact, and verifiably correct. The word "demonstrable" is important: in regulated industries and AI-driven workflows, fidelity that can be asserted is not the same as fidelity that can be proven. GoldenGate Veridata provides algorithmic verification of source-to-target fidelity rather than relying on assumption.

SQL_REDO
A reconstructed SQL statement that LogMiner generates to represent a committed change found in the redo log. SQL_REDO is a transformation of the original redo record, not a native representation of the committed transaction. For many simple DML operations this distinction is not material; for complex types, Oracle-specific constructs, and edge cases like rollbacks, the gap between reconstructed SQL and the actual committed operation can produce incorrect downstream results.

Supplemental Logging
Additional logging enabled in Oracle Database to ensure that redo records contain enough information for downstream change reconstruction and row identification. By default, Oracle's redo logs record only the minimum information needed for database recovery. Supplemental logging adds before-image columns, key columns, and other identifiers required by CDC tools to correctly identify which rows were changed and reconstruct complete change records. Most CDC tools require minimal supplemental logging at a minimum; some require all-column supplemental logging.

TDE (Transparent Data Encryption)
Oracle's database encryption technology that protects data at rest by encrypting tablespace content, columns, and redo log records. CDC tools that attempt to read redo logs from disk without Oracle API access encounter TDE-encrypted content they cannot decrypt without direct access to Oracle's key management infrastructure — creating a security architecture where CDC tool operators hold Master Encryption Key (MEK) access. GoldenGate's kernel-level integration handles TDE transparently, without requiring key exposure outside the Oracle security boundary.

Trail File
A GoldenGate file that stores captured change records in GoldenGate's native binary format for transport, buffering, and delivery to downstream processes. Trail files provide a durable, encrypted handoff point between Extract, routing processes, and Replicat. Because trail files persist committed changes independently of the source database's redo retention, they allow replication to absorb network outages, target unavailability, or processing delays without requiring source log retention beyond Oracle's own policies.

Transaction Boundary Integrity
The guarantee that all row changes belonging to a single source transaction are delivered downstream together as one coherent, atomic unit. A payment transaction that atomically updates an account balance, creates a ledger entry, and flags a risk position must arrive downstream as a single business event — not as three unrelated row changes scattered across delivery batches. CDC pipelines that do not preserve transaction boundaries allow downstream systems to observe partial states that the source database never exposed to any application.

VECTOR Datatype
An Oracle Database datatype introduced in Oracle AI Database 23ai for storing high-dimensional vector embeddings used by AI similarity search and Oracle AI Vector Search. CDC support for the VECTOR datatype is required when replicating AI-ready data pipelines that maintain vector indexes in sync with their source relational data. GoldenGate 26ai captures and replicates VECTOR data natively and supports inline embedding generation during replication via OCI Generative AI integration.

XStream
An Oracle Database API framework that enables client applications to receive change events from Oracle (XStream Out) or send changes into Oracle (XStream In). XStream Out delivers committed changes as Logical Change Records to connected clients and provides stronger ordering guarantees and performance than LogMiner. It is an API layer, not a complete replication platform: checkpointing, lifecycle management, schema handling, error recovery, and operational monitoring are the responsibility of the application or product built on top of XStream. See: Approach 2 — Oracle XStream APIs.

Appendix D: Citations and References

The following sources are referenced throughout this paper. Citations appear inline in IEEE numeric format, and the entries below resolve each bracketed number to its source.

[1] Oracle, "Managing the Redo Log," *Oracle AI Database Administrator's Guide, 26ai*. Accessed: Jun. 5, 2026. [Online]. Available: https://docs.oracle.com/en/database/oracle/oracle-database/26/admin/managing-the-redo-log.html

[2] Oracle, "Using LogMiner to Analyze Redo Log Files," *Oracle AI Database Utilities, 26ai*. Accessed: Jun. 5, 2026. [Online]. Available: https://docs.oracle.com/en/database/oracle/oracle-database/26/sutil/oracle-logminer-utility.html

[3] Oracle, "Oracle LogMiner," *Oracle Autonomous AI Database Serverless Documentation*. Accessed: Jun. 5, 2026. [Online]. Available: https://docs.public.content.oci.oraclecloud.com/en-us/iaas/autonomous-database-serverless/doc/autonomous-logminer.html

[4] Oracle, *Oracle GoldenGate Licensing Information, 26ai*, May 2026. Accessed: Jun. 5, 2026. [Online]. Available: https://docs.oracle.com/en/database/goldengate/core/26/ogglc/oracle-goldengate-licensing-guide.pdf

[5] Oracle, "Choosing Capture and Apply Modes," *Using Oracle GoldenGate for Oracle Database, 19.1*. Accessed: Jun. 5, 2026. [Online]. Available: https://docs.oracle.com/en/middleware/goldengate/core/19.1/oracle-db/choosing-different-apply-modes.html

[6] Oracle, "Configuring Capture in Integrated Mode," *Using Oracle GoldenGate for Oracle Database, 19.1*. Accessed: Jun. 5, 2026. [Online]. Available: https://docs.oracle.com/en/middleware/goldengate/core/19.1/oracle-db/configuring-extract.html

[7] Oracle, "Before Adding Extract and Replicat Processes," *Oracle GoldenGate Microservices Documentation, 26ai*. Accessed: Jun. 5, 2026. [Online]. Available: https://docs.oracle.com/en/database/goldengate/core/26/coredoc/configure-ogg-adding-extract-and-replicat.html

[8] Oracle, "ADD TRANDATA," *Oracle GoldenGate Command Line Interface Reference, 26ai*. Accessed: Jun. 5, 2026. [Online]. Available: https://docs.oracle.com/en/database/goldengate/core/26/gclir/add-trandata.html

[9] Oracle, "Prepare Database Connection, System, and Parameter Settings," *Oracle GoldenGate Microservices Documentation, 26ai*. Accessed: Jun. 5, 2026. [Online]. Available: https://docs.oracle.com/en/database/goldengate/core/26/coredoc/prepare-database-connection-system-and-parameter-settings-oracle.html

[10] Oracle, "XStream Out Concepts," *Oracle AI Database XStream Guide, 26ai*. Accessed: Jun. 5, 2026. [Online]. Available: https://docs.oracle.com/en/database/oracle/oracle-database/26/xstrm/xstream-out-concepts.html

[11] Snowflake, "About Openflow Connector for Oracle," *Snowflake Documentation*. Accessed: Jun. 5, 2026. [Online]. Available: https://docs.snowflake.com/en/user-guide/data-integration/openflow/connectors/oracle/about

[12]Snowflake, "Set Up Near Real-Time Oracle CDC to Snowflake Using Openflow," *Snowflake Developer Guides*. Accessed: Jun. 5, 2026. [Online]. Available: https://www.snowflake.com/en/developers/guides/near-real-time-oracle-cdc-to-snowflake-using-openflow/

[13]Confluent, "Oracle XStream CDC Source Connector for Confluent Cloud," *Confluent Documentation*. Accessed: Jun. 5, 2026. [Online]. Available: https://docs.confluent.io/cloud/current/connectors/cc-oracle-xstream-cdc-source/cc-oracle-xstream-cdc-source.html

[14]IBM, "Required Database, User Accounts, and Schemas," *IBM InfoSphere Data Replication Documentation*. Accessed: Jun. 5, 2026. [Online]. Available: https://www.ibm.com/docs/en/idr/11.4.0?topic=xstream-required-database-user-accounts-schemas

[15]Amazon Web Services, "Using Oracle GoldenGate with Amazon RDS for Oracle," *Amazon RDS User Guide*. Accessed: Jun. 5, 2026. [Online]. Available: https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Appendix.OracleGoldenGate.html

[16]Oracle, "New Features," *Oracle GoldenGate Release Notes, 26ai*. Accessed: Jun. 5, 2026. [Online]. Available: https://docs.oracle.com/en/database/goldengate/core/26/release-notes/new-features.html

[17]Oracle, "Understanding What's Supported," *Using Oracle GoldenGate for Oracle Database*. Accessed: Jun. 5, 2026. [Online]. Available: https://docs.oracle.com/en/middleware/goldengate/core/18.1/oracle-db/1-understanding-whats-supported.html

[18]Oracle, "Replicate Business Objects with Oracle JSON Relational Duality and GoldenGate Data Streams," *Oracle GoldenGate Microservices Documentation, 26ai*. Accessed: Jun. 5, 2026. [Online]. Available: https://docs.oracle.com/en/database/goldengate/core/26/coredoc/distribute-json-dv-ogg-data-streams.html

[19]Oracle, "Replicate Data Using Oracle GoldenGate," *Oracle Autonomous AI Database Documentation*. Accessed: Jun. 5, 2026. [Online]. Available: https://docs.oracle.com/en-us/iaas/autonomous-database/doc/use-oracle-goldengate-replicate-data-autonomous-database.html

[20]Oracle, "MAA Platinum to Achieve Never Down: Oracle GoldenGate and Oracle Database 19c," *Oracle Database High Availability Documentation*. Accessed: Jun. 5, 2026. [Online]. Available: https://docs.oracle.com/en/database/oracle/oracle-database/19/haiad/maa_platinum_ogg.html

[21]Oracle, "Oracle GoldenGate Security Features," *Oracle GoldenGate Security Guide, 26ai*. Accessed: Jun. 5, 2026. [Online]. Available: https://docs.oracle.com/en/database/goldengate/core/26/coredoc/secure-standards.html

[22]Oracle, "Secure Data at Rest," *Oracle GoldenGate Security Guide, 26ai*. Accessed: Jun. 5, 2026. [Online]. Available: https://docs.oracle.com/en/database/goldengate/core/26/coredoc/secure-data-rest.html

[23]Oracle, "Secure Data in Transit," *Oracle GoldenGate Security Guide, 26ai*. Accessed: Jun. 5, 2026. [Online]. Available: https://docs.oracle.com/en/database/goldengate/core/26/coredoc/secure-data-transit.html

[24]Oracle, “GoldenGate Supported System Configurations,” *Oracle GoldenGate Certifications*. Accessed: Jun. 5, 2026. [Online]. Available: https://www.oracle.com/integration/goldengate/certifications/

[25]Oracle, “Introduction to Oracle GoldenGate Veridata,” *Administering Oracle GoldenGate Veridata*. Accessed: Jun. 5, 2026. [Online]. Available: https://docs.oracle.com/goldengate/v12212/gg-veridata/GVDAD/introduction-oracle-goldengate-veridata.htm

[26]Oracle, “Risks of Using Binary Log Readers with Oracle Database,” *Oracle Data Integration Blog*, Oct. 2025. Accessed: Jun. 5, 2026. [Online]. Available: https://blogs.oracle.com/dataintegration/binary-log-readers

[27]Oracle, *Transactional Oracle Master Agreement (TOMA)*. Accessed: Jun. 5, 2026. [Online]. Available: https://www.oracle.com/contracts/docs/lic_online_toma_for_o.com_v062223_ch_ca_eng.pdf

[28]Oracle, *Announced at Oracle AI World New York, March 2026 in forum presentation*

ORACLE

ORACLE CORPORATION

Worldwide Headquarters
Austin, TX 78741 USA

Worldwide Inquiries
TELE + 1.650.506.7000 + 1.800.ORACLE1
FAX + 1.650.506.7200
oracle.com

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/oracle facebook.com/oracle x.com/oracle

Copyright © 2026, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0626

Summer 2026
Author: Oracle Product Development