

Oracle GoldenGate - Limitations and Restrictions

This page describes which Oracle data types are supported by Oracle GoldenGate. It is intended to help determine whether classic or integrated capture is the more appropriate mode to replicate a specific database. The decision will largely be determined support limitations and restrictions for the tables that need to be replicated.

Classic capture will probably be suitable for most databases. Integrated capture is appropriate for databases with tables containing columns of advanced data types such as LOBs and XMLType.

This page attempts to summarize the Oracle GoldenGate Oracle Installation and Setup Guide Release 11.2.1 (E35957-01). For complete and current details please refer to this document.

At the time of writing the current Oracle GoldenGate version was 11.2.1.0.1

Introduction

Oracle GoldenGate supports two types of capture:

- Classic Capture
- Integrated Capture

Classic capture is a recent name for the original GoldenGate capture mechanism which reads data directly from the online redo logs and/or archived redo logs where possible. Additional data may be fetched from the database file where necessary.

Integrated capture was introduced in Oracle GoldenGate 11.2.1. It was initially available for Oracle 11.2.0.3 with the 11.2.0.3 Database specific bundle patch for Integrated Extract 11.2.x (MOS Note 1411356.1)

Integrated capture uses a log mining server on the source system or in a downstream Oracle database, similar to a Data Guard logical standby or Oracle Streams.

Where possible, data is captured from the redo log. However, for some data types the redo data is incomplete and it is necessary to fetch additional data from the database. Where additional data is fetched as part of a separate transaction, there is a possibility of inconsistency

Data Type Support

Scalar Data Types

The following scalar data types are supported for both classic and integrated capture:

- NUMBER
- BINARY FLOAT
- BINARY DOUBLE
- CHAR
- VARCHAR2
- LONG
- NCHAR
- NVARCHAR2
- RAW
- LONG RAW
- DATE
- TIMESTAMP

Scalar columns are captured from the redo by both classic capture and integrated capture.

Date and Timestamp

DATE and TIMESTAMP columns are fully supported.

There is limited support for:

- INTERVAL DAY
- INTERVAL YEAR
- TIMESTAMP WITH TIME ZONE
- TIMESTAMP WITH LOCAL TIME ZONE

Large Object Types (LOBs)

The following large object types are supported in both classic and integrated capture modes:

- CLOB
- NCLOB
- BLOB

BASICFILE and SECUREFILE LOBs are both supported

For both classic and integrated capture modes:

- For LOBs exceeding 4K in size, it is stored in segments in the GoldenGate trail. The first 4K is stored in the base segment; the remainder is stored in 2K segments.
- For LOBs exceeding 4K in size, filtering, column mapping or manipulation is not supported

For classic capture mode only BASICFILE LOBs are fetched from the database when:

- Extract determines the LOB is invalid
- LOB is created with NOLOGGING option
- LOB is created with CACHE attribute
- LOB is only partially updated using OCI calls or DBMS_LOB package

For classic capture mode only SECUREFILE LOBs are captured from redo logs only when:

- Update is complete
- LOB is not transformed (compressed, encrypted or deduplicated)
- LOB is stored out-of-row

For classic capture mode only SECUREFILE LOBs are fetched from the database when:

- Extract determines LOB is invalid
- LOB is stored in-row
- LOB is transformed with compression or encryption
- LOB is created with CACHE attribute
- LOB data is missing from redo log (NO_LOGGING or deduplicate)
- LOB is partially updated using OCI calls or DBMS_LOB package

XML Types

The following XML types are supported in both classic and integrated capture mode:

- XMLType columns
- XMLType tables stored as:
 - XML CLOB
 - XML Binary

In addition XMLType tables stored as XML Object Relational are supported for integrated capture mode only.

In both classic and integrated capture modes:

- XMLType data is treated as a LOB
- Source and target objects containing XML must be identical
- Filtering and manipulation are not supported
- XML objects can be mapped to character columns using COLMAP clause in TABLE and MAP statements
- The following are not supported:
- Hierarchy-enabled tables
 - XMLType tables created from a CTAS statement
 - XMLType tables with primary key-based object identifiers (OID)
 - Non-XMLType tables with a single XML column
 - SQL*Loader direct path insert for XML Binary and XML Object Relational
- XMLSchema-based XMLType tables and columns are supported:
 - Changes to XMLSchemas are not replicated - must be registered on both source and target databases using DBMS_XML package
- Supported tables must have a unique key constraint or a set of scalar columns that ensure uniqueness
- Primary key constraints containing XML attributes cannot be used for row identification

In integrated capture mode the following additional limitations apply:

- Oracle database COMPATIBLE parameter must be 11.2.0.3.0 or above
- Maximum length for SET value of an update to an XMLType is 32K.

In classic capture mode the following additional limitations apply:

- For XML Binary, additional row data is fetched from the source database because redo log does not contain enough information - can lead to inconsistency.
- Tables containing XMLType columns must have primary key, unique index or unique constraint

User Defined Types

User-defined types are supported in both classic and integrated capture modes:

- Source and target objects containing UDTs must have the same structure. Schema names can differ
- UDTs are fetched from the database. Tables containing UDTs must have a primary key, unique index or unique constraint
- UDTs containing the following types are not supported:
 - CLOB
 - CFILE
 - BFILE
 - INTERVAL_YM
 - INTERVAL_DS
 - OPAQUE (except XMLType)
- GoldenGate RMTTASK parameter does not support UDTs
- CHAR and VARCHAR attributes containing binary or unprintable characters are not supported.
- REF types are not supported

Collection Types

VARRAYs and nested tables are supported in both classic and integrated capture modes:

- When data in a nested table is updated, the row containing the nested table must also be updated at the same time
- When VARRAYs and nested tables are fetched, the entire contents of the column are fetched each time, not just the changes

Object Tables

Object tables are supported in both classic and integrated capture modes:

- Supported in both uni-directional and bi-directional configurations
- Object tables are captured from the redo log
 - Some object table columns including LOBs and collection types are fetched from the database
- Object tables can be mapped to non-Oracle object tables
- Object tables with key containing UDT are not supported
- Object tables where UDT is only column are not supported
- A primary key must be defined on the root-level object attributes of the object table
 - Cannot include leaf-level attributes
 - If no key defined, all viable columns will be used as a pseudo-key
- Replication of DDL operations for object tables is not supported
 - Includes object versioning
- Synonyms not supported for object tables and for relational tables containing objects

Spatial Types

The following spatial types are supported in both classic and integrated capture modes:

- SDO_GEOMETRY
- SDO_TOPO_GEOMETRY
- SDO_GEORASTER

Additional configuration is required

Other Data Types

The following data types are not supported:

- Abstract data types with scalar, LOBs, VARRAYS, nested tables and/or REFs
- ANYDATA
- ANYDATASET
- ANYTYPE
- BFILE
- MLSLABEL
- ORDDICOM
- TIMEZONE_ABBR
- URITYPE
- UROWID

DML Support

DML is supported to:

- Regular tables,
- Index-organized tables
- Clustered tables
- Materialized views.

The following operations are supported:

- INSERT
- UPDATE
- DELETE
- Associated transaction control operations

The following are not supported in either classic or integrated capture mode:

- Database replay
- EXTERNAL tables

The following are not supported in classic capture mode:

- Synonyms

Single Column Tables

One column tables are supported except where the column is one of the following data types:

- LOB
- LONG
- Nested Table
- User defined data type
- VARRAY
- XML

Unused Columns

Support of tables with unused columns are supported

- Support is disabled by default.
- Extract will abend by default
- Support must be enabled using ALLOWUNUSEDCOLUMN option

Partitioning

The following types of partitioning are supported in both classic and integrated capture modes:

- Range partitioning
- Hash partitioning
- List partitioning
- Interval partitioning
- System partitioning
- Composite partitioning
- Reference partitioning
- Virtual column-based partitioning

Virtual Columns

Tables with virtual columns are supported. Data is not captured from or applied to virtual columns

Changes to virtual columns are not logged in the online redo log and therefore cannot be extracted by GoldenGate.

Initial load data cannot be applied to a virtual column.

Virtual source columns can be mapped to non-virtual target columns

Virtual columns in unique keys and indexes are ignored

If a virtual column is the only unique identifier for a table, the remaining columns will be used for row identification. This can lead to table corruption if the remaining columns do not ensure uniqueness.

Transparent Data Encryption

TDE is supported in both classic and integrated capture modes

For integrated capture, source database must be 11.1 or above with COMPATIBLE parameter set to 11.0.0.0 or above.

- Column-level encryption is supported for Oracle 10.2.0.5 and above
- Tablespace-level encryption is supported for Oracle 10.2.0.5, 11.2.0.1 and above

For classic capture, TDE requires some additional configuration.

TRUNCATE Statements

TRUNCATE statements are supported in both classic and integrated capture modes as either:

- part of DDL replication
- standalone functionality independent of DDL replication

Distributed Transactions

The following are not supported in classic capture mode:

- Distributed transactions
- XA and PDML distributed transactions

In Oracle 11.1.0.6 and above distributed transactions can be captured if they non-distributed transactions have been configured at database level by setting the static `_CLUSTERWIDE_GLOBAL_TRANSACTIONS` database parameter to FALSE

Compression

The following are not supported in classic capture mode:

- Capture from tables using OLTP table compression
- Capture from tablespaces and tables created or altered with COMPRESS
- Exadata Hybrid Columnar Compression

Direct-Load Inserts

Capture of direct-load inserts is supported in both classic and integrated capture modes.

The following direct-load methods are supported:

- APPEND hint
- PARALLEL hint
- SQLLDR with DIRECT=TRUE

In classic capture mode the PARALLEL hint is only supported for non-RAC databases

Index Organized Tables

In both classic and integrated capture modes the following limitations apply:

- DDL for an IOT with mapping table will be replicated correctly, but subsequent DML on the IOT will fail.

In classic capture mode the following limitations apply:

- IOT with a mapping table is not supported

- IOTs with prefix compression specified by the COMPRESS clause are not supported
- Direct load inserts into IOTs cannot specify the SORTED clause

Views

Views are supported in both classic capture and integrated capture modes with the following limitations:

- Capture from a view is supported when Extract is in initial-load mode (capture directly from source view, not the redo log)
- Change data is not captured from a view, but it can be captured from the underlying tables of the view
- View replication is only supported for inherently updateable views in which case the source table and target view structures must be identical.

Materialized Views

Materialized views are supported in both classic capture and integrated capture modes with the following limitations:

- Materialized views created WITH ROWID are not supported
- Materialized view logs can be created WITH ROWID
- Source table must have a primary key
- Truncation of materialized views is not supported. DELETE statements can be used.
- Some GoldenGate load methods do not support LOBs in a materialized view
- For Replicat the materialized view must be updatable
- DML from a full refresh of materialized view is supported; DDL is not

Clustered Tables

Index clusters and hash clusters are both supported in classic capture and integrated capture mode.

In classic capture mode, the following limitations apply:

- Encrypted and compressed clustered tables are not supported
- DML changes are captured while cluster size is unchanged. Any DDL that increases or decreases cluster size may cause subsequent GoldenGate extracts to fail.

Sequences

Active-Passive (uni-directional) - replication of sequence values is supported for both classic capture and integrated capture modes

- CACHE and INCREMENT BY settings on source and target database must be identical. NOCACHE can be specified
- Sequence can be CYCLE or NOCYCLE. Setting must be identical on both source and target.
- GoldenGate ensures target sequence values will always be higher than those of the source (or equal if CACHE is 0)

Active-Active (bi-directional) replication of sequence values is not supported for either mode

Oracle-Reserved Schemas

The following schema names are reserved by Oracle and should be excluded from GoldenGate replication:

\$AURORA	DBSNMP	ORDPLUGINS	SYSMAN
\$JIS	DMSYS	ORDSYS	SYSTEM
\$ORB	DSSYS	OSE\$HTTP\$ADMIN	TRACESVR
\$UNAUTHENTICATED	EXFSYS	OUTLN	WKPROXY

\$UTILITY	MDSYS	PERFSTAT	WKSYS
ANONYMOUS	ODM	PUBLIC	WMSYS
AURORA	ODM_MTR	REPADMIN	XDB
CTXSYS	OLAPSYS	SYS	

DDL Support

DDL is supported as follows for both classic capture and integrated capture modes.

DDL is supported for all Oracle GoldenGate topology configurations

DDL operations up to 2MB in size are supported on the following objects:

Clusters	Procedures	Tablespaces
Functions	Roles	Triggers
Indexes	Sequences	Types
Materialized Views	Synonyms	Users
Packages	Tables	Views

The 2MB size limitation includes packages, procedures and functions.

Active-active (bi-directional) replication of DDL is supported between two (and only two) databases that contain identical metadata.

The following DDL constructs are not supported by either capture method:

- ALTER TABLE MOVE TABLESPACE
- ALTER DATABASE
- ALTER SYSTEM
- DDL on nested tables
- DDL on a standby database

In addition classic capture mode does not support DDL involving password-based column encryption e.g.

```
CREATE TABLE t1 (c1 NUMBER,c2 VARCHAR2(32) ENCRYPT IDENTIFIED BY password);
ALTER TABLE t1 ADD COLUMN (c3 VARCHAR2(32) ENCRYPT IDENTIFIED BY password);
```