45

# DBMS\_CAPTURE\_ADM

The DBMS\_CAPTURE\_ADM package, one of a set of Oracle Replication packages, provides subprograms for starting, stopping, and configuring a capture process. The source of the captured changes is the redo logs, and the repository for the captured changes is a queue.



A multitenant container database is the only supported architecture in Oracle Database 21c and later releases. While the documentation is being revised, legacy terminology may persist. In most cases, "database" and "non-CDB" refer to a CDB or PDB, depending on context. In some contexts, such as upgrades, "non-CDB" refers to a non-CDB from a previous release.

This chapter contains the following topics:

- Overview
- Security Model
- Summary of DBMS CAPTURE ADM Subprograms

# DBMS CAPTURE ADM Overview

This package provides interfaces to start, stop, and configure a capture process or a synchronous capture. This package includes subprograms for preparing database objects for instantiation.

Capture processes can be used in an XStream configuration in a multitenant container database (CDB). A CDB is an Oracle database that includes zero, one, or many user-created pluggable databases (PDBs).



Oracle Database Concepts for more information about CDBs and PDBs

# DBMS\_CAPTURE\_ADM Security Model

The DBMS CAPTURE ADM security can be controlled in one of two ways.

- Granting EXECUTE on this package to selected users or roles.
- Granting EXECUTE CATALOG ROLE to selected users or roles.

If subprograms in the package are run from within a stored procedure, then the user who runs the subprograms must be granted EXECUTE privilege on the package directly. It cannot be granted through a role.

When the DBMS\_CAPTURE\_ADM package is used to manage an Oracle Replication configuration, it requires that the user is granted the privileges of an Oracle Replication administrator.

When the DBMS\_CAPTURE\_ADM package is used to manage an XStream configuration, it requires that the user is granted the privileges of an XStream administrator.

### Note:

- The user must be granted additional privileges to perform some administrative tasks using the subprograms in this package, such as setting a capture user. If additional privileges are required for a subprogram, then the privileges are documented in the section that describes the subprogram.
- Using XStream requires purchasing a license for the Oracle GoldenGate product. See *Oracle Database XStream Guide*.

## See Also:

Oracle Database XStream Guide for information about configuring an XStream administrator

# Summary of DBMS\_CAPTURE\_ADM Subprograms

This table lists the DBMS CAPTURE ADM subprograms and briefly describes them.

Table 45-1 DBMS\_CAPTURE\_ADM Package Subprograms

Subprogram	Description
ABORT_GLOBAL_INSTANTIATION Procedure	Reverses the effects of running the PREPARE_GLOBAL_INSTANTIATION, PREPARE_SCHEMA_INSTANTIATION, and PREPARE_TABLE_INSTANTIATION procedures
ABORT_SCHEMA_INSTANTIATION Procedure	Reverses the effects of running the PREPARE_SCHEMA_INSTANTIATION and PREPARE_TABLE_INSTANTIATION procedures
ABORT_SYNC_INSTANTIATION Procedure	Reverses the effects of running the PREPARE_SYNC_INSTANTIATION procedure
ABORT_TABLE_INSTANTIATION Procedure	Reverses the effects of running the PREPARE_TABLE_INSTANTIATION procedure
ALTER_CAPTURE Procedure	Alters a capture process
ALTER_SYNC_CAPTURE Procedure	Alters a synchronous capture



Table 45-1 (Cont.) DBMS\_CAPTURE\_ADM Package Subprograms

Subprogram	Description
BUILD Procedure	Extracts the data dictionary of the current database to the redo logs and automatically specifies database supplemental logging for all primary key and unique key columns
CREATE_CAPTURE Procedure	Creates a capture process
CREATE_SYNC_CAPTURE Procedure	Creates a synchronous capture
DROP_CAPTURE Procedure	Drops a capture process
INCLUDE_EXTRA_ATTRIBUTE Procedure	Includes or excludes an extra attribute in logical change records (LCRs) captured by the specified capture process or synchronous capture
PREPARE_GLOBAL_INSTANTIATION Procedure	Performs the synchronization necessary for instantiating all the tables in the database at another database and can enable supplemental logging for key columns or all columns in these tables
PREPARE_SCHEMA_INSTANTIATION Procedure	Performs the synchronization necessary for instantiating all tables in the schema at another database and can enable supplemental logging for key columns or all columns in these tables
PREPARE_SYNC_INSTANTIATION Function	Performs the synchronization necessary for instantiating one or more tables at another database and returns the prepare SCN
PREPARE_TABLE_INSTANTIATION Procedure	Performs the synchronization necessary for instantiating the table at another database and can enable supplemental logging for key columns or all columns in the table
SET_PARAMETER Procedure	Sets a capture process parameter to the specified value
START_CAPTURE Procedure	Starts the capture process, which mines redo logs and enqueues the mined redo information into the associated queue
STOP_CAPTURE Procedure	Stops the capture process from mining redo logs



All subprograms commit unless specified otherwise.

# ABORT\_GLOBAL\_INSTANTIATION Procedure

This procedure reverses the effects of running the PREPARE\_GLOBAL\_INSTANTIATION, PREPARE\_SCHEMA\_INSTANTIATION, and PREPARE\_TABLE\_INSTANTIATION procedures.

Specifically, this procedure performs the following actions:

- Removes data dictionary information related to the database, schema, and table instantiations
- Removes any supplemental logging enabled by the PREPARE\_GLOBAL\_INSTANTIATION, PREPARE SCHEMA INSTANTIATION, and PREPARE TABLE INSTANTIATION procedures

#### **Syntax**

#### **Parameter**

Table 45-2 ABORT\_GLOBAL\_INSTANTIATION Procedure Parameter

Parameter	Description
container	Either CURRENT , ALL, or pdb_name.
	If CURRENT is specified, then this procedure removes supplemental logging for the current container.
	If ALL is specified, then this procedure removes supplemental logging for all of the containers in the current CDB.
	If pdb_name is specified, then this procedure removes supplemental logging for the specified PDB.
	$\mathtt{ALL}$ and $\mathtt{pdb\_name}$ are valid only when you invoke the procedure from the root.

# ABORT\_SCHEMA\_INSTANTIATION Procedure

This procedure reverses the effects of running the PREPARE\_SCHEMA\_INSTANTIATION procedure. It also reverses the effects of running the PREPARE\_TABLE\_INSTANTIATION procedure on tables in the specified schema.

Specifically, this procedure performs the following actions:

- Removes data dictionary information related to schema instantiations and table instantiations of tables in the schema
- Removes any supplemental logging enabled by the PREPARE\_SCHEMA\_INSTANTIATION procedure
- Removes any supplemental logging enabled by the PREPARE\_TABLE\_INSTANTIATION
  procedure for tables in the specified schema

### **Syntax**

#### **Parameter**

### Table 45-3 ABORT\_SCHEMA\_INSTANTIATION Procedure Parameter

Parameter	Description
schema_name	The name of the schema for which to abort the effects of preparing instantiation



Table 45-3 (Cont.) ABORT\_SCHEMA\_INSTANTIATION Procedure Parameter

Parameter	Description
container	Either CURRENT, ALL, or pdb_name.
	If CURRENT is specified, then this procedure removes supplemental logging for the current container.
	If ${\tt ALL}$ is specified, then this procedure removes supplemental logging for all of the containers in the current CDB.
	If pdb_name is specified, then this procedure removes supplemental logging for the specified PDB.
	ALL and pdb_name are valid only when you invoke the procedure from the root.

# ABORT\_SYNC\_INSTANTIATION Procedure

This procedure reverses the effects of running the PREPARE\_SYNC\_INSTANTIATION procedure. Specifically, this procedure removes data dictionary information related to the table instantiation.

This procedure is overloaded. The table\_names parameter is VARCHAR2 datatype in one version and DBMS UTILITY.UNCL ARRAY datatype in the other version.

### **Syntax**

```
DBMS_CAPTURE_ADM.ABORT_SYNC_INSTANTIATION(
   table_names IN VARCHAR2);

DBMS_CAPTURE_ADM.ABORT_SYNC_INSTANTIATION(
   table names IN DBMS_UTILITY.UNCL_ARRAY);
```

#### **Parameters**

Table 45-4 ABORT\_SYNC\_INSTANTIATION Procedure Parameter

Parameter	Description
table_names	When the table_names parameter is VARCHAR2 datatype, a comma-delimited list of the tables for which to abort the effects of preparing instantiation. There must be no spaces between entries.
	When the table_names parameter is DBMS_UTILITY.UNCL_ARRAY datatype, specify a PL/SQL associative array of this type that contains the names of the tables for which to abort the effects of preparing instantiation. The first table name is at position 1, the second at position 2, and so on. The table does not need to be NULL terminated.
	In either version of the procedure, specify the name of each table in the form [schema_name.] table_name. For example, hr.employees. If the schema is not specified, then the current user is the default.

# ABORT\_TABLE\_INSTANTIATION Procedure

This procedure reverses the effects of running the PREPARE TABLE INSTANTIATION procedure.

Specifically, this procedure performs the following actions:

Removes data dictionary information related to the table instantiation

Removes any supplemental logging enabled by the PREPARE\_TABLE\_INSTANTIATION procedure

### **Syntax**

```
DBMS_CAPTURE_ADM.ABORT_TABLE_INSTANTIATION(
  table_name IN VARCHAR2);
  container IN VARCHAR2 DEFAULT 'CURRENT');
```

#### **Parameter**

Table 45-5 ABORT\_TABLE\_INSTANTIATION Procedure Parameter

Parameter	Description
Parameter	Description
table_name	The name of the table for which to abort the effects of preparing instantiation, specified as [schema_name.]object_name. For example, hr.employees. If the schema is not specified, then the current user is the default.
container	Either CURRENT, ALL, or pdb_name.
	If CURRENT is specified, then this procedure removes supplemental logging for the current container.
	If ALL is specified, then this procedure removes supplemental logging for all of the containers in the current CDB.
	If pdb_name is specified, then this procedure removes supplemental logging for the specified PDB.
	ALL and pdb_name are valid only when you invoke the procedure from the root.

# ALTER\_CAPTURE Procedure

This procedure alters a capture process.

#### **Syntax**

Table 45-6 ALTER CAPTURE Procedure Parameters

Parameter	Description
capture_name	The name of the capture process being altered. You must specify an existing capture process name. Do not specify an owner.



Table 45-6 (Cont.) ALTER\_CAPTURE Procedure Parameters

Parameter	Description
rule_set_name	The name of the positive rule set for the capture process. The positive rule set contains the rules that instruct the capture process to capture changes.
	To change the positive rule set for the capture process, specify an existing rule set in the form [schema_name.]rule_set_name. For example, to specify a positive rule set in the hr schema named job_capture_rules, enter hr.job_capture_rules. If the schema is not specified, then the current user is the default.
	An error is returned if the specified rule set does not exist. You can create a rule set and add rules to it using the DBMS_RULE_ADM package.
	If you specify <code>NULL</code> and the <code>remove_rule_set</code> parameter is set to <code>FALSE</code> , then the procedure retains any existing positive rule set. If you specify <code>NULL</code> and the <code>remove_rule_set</code> parameter is set to <code>TRUE</code> , then the procedure removes any existing positive rule set.
remove_rule_set	If TRUE, then the procedure removes the positive rule set for the specified capture process. If you remove a positive rule set for a capture process, and the capture process does not have a negative rule set, then the capture process captures all supported changes to all objects in the database, excluding database objects in the SYS and SYSTEM schemas.
	If you remove a positive rule set for a capture process, and the capture process has a negative rule set, then the capture process captures all supported changes that are not discarded by the negative rule set.
	If FALSE, then the procedure retains the positive rule set for the specified capture process.
	If the ${\tt rule\_set\_name}$ parameter is non-NULL, then ensure that this parameter is set to <code>FALSE</code> .
start_scn	A valid SCN for the database from which the capture process starts capturing changes. The SCN value must be greater than or equal to the first SCN for the capture process. Also, the capture process must be stopped before resetting its start SCN. An error is returned if an invalid SCN is specified or if the capture process is enabled.



Table 45-6 (Cont.) ALTER\_CAPTURE Procedure Parameters

#### Parameter

#### Description

use database link

If TRUE, then the capture process at a downstream database uses a database link to the source database for administrative purposes relating to the capture process. If you want a capture process that is not using a database link currently to begin using a database link, then specify TRUE. In this case, a database link with the same name as the global name of the source database must exist at the downstream database.

If FALSE, then either the capture process is running on the source database, or the capture process at a downstream database does not use a database link to the source database. If you want a capture process that is using a database link currently to stop using a database link, then specify FALSE. In this case, you must prepare source database objects for instantiation manually when you add or change capture process rules that pertain to these objects.

If  ${\tt NULL},$  then the current value of this parameter for the capture process is not changed.

The lowest SCN in the redo log from which a capture process can capture changes. If you specify a new first SCN for the capture process, then the specified first SCN must meet the following requirements:

- It must be greater than the current first SCN for the capture process.
- It must be less than or equal to the current applied SCN for the capture process. However, this requirement does not apply if the current applied SCN for the capture process is zero.
- It must be less than or equal to the required checkpoint SCN for the capture process.

An error is returned if the specified SCN does not meet the first three requirements. See "Usage Notes" for information about determining an SCN value that meets all of these conditions.

When the first SCN is modified, the capture process purges information from its LogMiner data dictionary that is required to restart it at an earlier SCN. See BUILD Procedure for more information about a LogMiner data dictionary.

If the specified first SCN is higher than the current start SCN for the capture process, then the start SCN is set automatically to the new value of the first SCN.

first scn



Table 45-6 (Cont.) ALTER\_CAPTURE Procedure Parameters

#### Parameter

### Description

negative rule set name

The name of the negative rule set for the capture process. The negative rule set contains the rules that instruct the capture process to discard changes.

To change the negative rule set for the capture process, specify an existing rule set in the form

 $[schema\_name.] \ rule\_set\_name. \ For example, to specify a negative rule set in the $hr$ schema named$ 

neg\_capture\_rules, enter hr.neg\_capture\_rules. If the schema is not specified, then the current user is the default.

An error is returned if the specified rule set does not exist. You can create a rule set and add rules to it using the DBMS RULE ADM package.

If you specify <code>NULL</code> and the <code>remove\_negative\_rule\_set</code> parameter is set to <code>FALSE</code>, then the procedure retains any existing negative <code>rule\_set</code> lf you specify <code>NULL</code> and the <code>remove\_negative\_rule\_set</code> parameter is set to <code>TRUE</code>, then the procedure removes any existing negative <code>rule\_set</code>.

If you specify both a positive and a negative rule set for a capture process, then the negative rule set is always evaluated first.

remove negative rule set

If TRUE, then the procedure removes the negative rule set for the specified capture process. If you remove a negative rule set for a capture process, and the capture process does not have a positive rule set, then the capture process captures all supported changes to all objects in the database, excluding database objects in the SYS and SYSTEM schemas.

If you remove a negative rule set for a capture process, and a positive rule set exists for the capture process, then the capture process captures all changes that are not discarded by the positive rule set.

If  ${\tt FALSE},$  then the procedure retains the negative rule set for the specified capture process.

If the negative\_rule\_set\_name parameter is non-NULL, then ensure that this parameter is set to FALSE.



Table 45-6 (Cont.) ALTER\_CAPTURE Procedure Parameters

Parameter	Description
capture_user	The user in whose security domain a capture process captures changes that satisfy its rule sets and runs custom rule-based transformations configured for capture process rules. If NULL, then the capture user is not changed.
	To change the capture user, the user who invokes the ALTER_CAPTURE procedure must be granted the DBA role. Only the SYS user can set the capture_user to SYS.
	If you change the capture user, then this procedure grants the new capture user enqueue privilege on the queue used by the capture process and configures the user as a secure queue user of the queue. In addition, ensure that the capture user has the following privileges:
	<ul> <li>EXECUTE privilege on the rule sets used by the capture process</li> </ul>
	<ul> <li>EXECUTE privilege on all rule-based transformation functions used in the rule set</li> </ul>
	These privileges can be granted directly to the capture user, or they can be granted through roles.
	In addition, the capture user must be granted EXECUTE privilege on all packages, including Oracle-supplied packages, that are invoked in rule-based transformations run by the capture process. These privileges must be granted directly to the capture user. They cannot be granted through roles.
	The capture process is stopped and restarted automatically when you change the value of this parameter.
	<b>Note:</b> If the capture user for a capture process is dropped using DROP USER CASCADE, then the capture process is also dropped automatically.
<pre>checkpoint_retention_time</pre>	Either the number of days that a capture process retains checkpoints before purging them automatically, or DBMS_CAPTURE_ADM. INFINITE if checkpoints should not be purged automatically. If NULL, then the checkpoint retention time is not changed.
	If a number is specified, then a capture process purges a checkpoint the specified number of days after the checkpoint was taken. Partial days can be specified using decimal values. For example, .25 specifies 6 hours.
	When a checkpoint is purged, LogMiner data dictionary information for the archived redo log file that corresponds to the checkpoint is purged, and the first_scn of the capture process is reset to the SCN value corresponding to the first change in the next archived redo log file.
start_time	A valid time from which the capture process starts capturing changes. The capture process must be stopped before resetting its start time.
	An error is returned if an invalid time is specified or if the capture process is enabled.
	The start_scn and start_time parameters are mutually exclusive.
oldest_scn	The oldest SCN of the transactions currently being processed.



If you want to alter the first SCN for a capture process, then the value specified must meet the conditions in the description for the first scn parameter.

### **Examples**

The following query determines the current first SCN, applied SCN, and required checkpoint SCN for each capture process in a database:

```
SELECT CAPTURE_NAME, FIRST_SCN, APPLIED_SCN, REQUIRED_CHECKPOINT_SCN FROM DBA_CAPTURE;
```

# ALTER\_SYNC\_CAPTURE Procedure

This procedure alters a synchronous capture.

#### **Syntax**

#### **Parameters**

### Table 45-7 ALTER\_SYNC\_CAPTURE Procedure Parameters

Parameter	Description
capture_name	The name of the synchronous capture being altered. You must specify an existing synchronous capture name. Do not specify an owner.
rule_set_name	The name of the positive rule set for the synchronous capture. The positive rule set contains the rules that instruct the synchronous capture to capture changes.
	To change the rule set for the synchronous capture, specify an existing rule set in the form [schema_name.]rule_set_name. For example, to specify a positive rule set in the strmadmin schema named sync_cap_rules, enter strmadmin.sync_cap_rules. If the schema is not specified, then the current user is the default.  An error is returned if the specified rule set does not exist.  If NULL, then the rule set is not changed.



Table 45-7 (Cont.) ALTER\_SYNC\_CAPTURE Procedure Parameters

#### Parameter

#### Description

capture user

The user in whose security domain a synchronous capture captures changes that satisfy its rule set and runs custom rule-based transformations configured for synchronous capture rules. If NULL, then the capture user is not changed.

To change the capture user, the user who invokes the <code>ALTER\_SYNC\_CAPTURE</code> procedure must be granted the <code>DBA</code> role. Only the <code>SYS</code> user can set the <code>capture user to SYS</code>.

If you change the capture user, then this procedure grants the new capture user enqueue privilege on the queue used by the synchronous capture and configures the user as a secure queue user of the queue. In addition, ensure that capture user has the following privileges:

- EXECUTE privilege on the rule sets used by the synchronous capture
- EXECUTE privilege on all rule-based transformation functions used in the rule set

These privileges can be granted directly to the capture user, or they can be granted through roles.

In addition, the capture user must be granted EXECUTE privilege on all packages, including Oracle-supplied packages, that are invoked in rule-based transformations run by the synchronous capture. These privileges must be granted directly to the capture user. They cannot be granted through roles.

### **Usage Notes**

If the capture user for a synchronous capture is dropped using DROP USER . . . CASCADE, then the synchronous capture is also dropped automatically.

## **BUILD Procedure**

This procedure extracts the data dictionary of the current database to the redo log and automatically specifies database supplemental logging by running the SQL statement ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;



A multitenant container database is the only supported architecture in Oracle Database 21c and later releases. While the documentation is being revised, legacy terminology may persist. In most cases, "database" and "non-CDB" refer to a CDB or PDB, depending on context. In some contexts, such as upgrades, "non-CDB" refers to a non-CDB from a previous release.

This procedure is overloaded. One version of this procedure contains the  $\mathtt{OUT}$  parameter first  $\mathtt{scn}$ , and the other does not.

#### **Syntax**

```
DBMS_CAPTURE_ADM.BUILD(
   first_scn OUT NUMBER);
DBMS CAPTURE ADM.BUILD;
```



#### **Parameters**

Table 45-8 BUILD Procedure Parameter

Parameter	Description
first_scn	Contains the lowest SCN value corresponding to the data dictionary extracted to the redo log that can be specified as a first SCN for a capture process

#### **Usage Notes**

The following usage notes apply to this procedure:

- You can run this procedure multiple times at a source database.
- If you plan to capture changes originating at a source database with a capture process, then this procedure must be executed at the source database at least once. When the capture process is started, either at a local source database or at a downstream database, the capture process uses the extracted information in the redo log to create a LogMiner data dictionary.
- A LogMiner data dictionary is a separate data dictionary used by a capture process to
  determine the details of a change that it is capturing. The LogMiner data dictionary is
  necessary because the primary data dictionary of the source database might not be
  synchronized with the redo data being scanned by a capture process.
- After executing this procedure, you can query the FIRST\_CHANGE# column of the V\$ARCHIVED\_LOG dynamic performance view where the DICTIONARY\_BEGIN column is YES to determine the lowest SCN value for the database that can be specified as a first SCN for a capture process. The first SCN for a capture process is the lowest SCN in the redo log from which the capture process can capture changes. You can specify the first SCN for a capture process when you run the CREATE\_CAPTURE or ALTER\_CAPTURE procedure in the DBMS\_CAPTURE\_ADM package.
- In a CDB, the BUILD procedure must be executed from the root.

# CREATE CAPTURE Procedure

This procedure creates a capture process.



A multitenant container database is the only supported architecture in Oracle Database 21c and later releases. While the documentation is being revised, legacy terminology may persist. In most cases, "database" and "non-CDB" refer to a CDB or PDB, depending on context. In some contexts, such as upgrades, "non-CDB" refers to a non-CDB from a previous release.

## See Also:

DBMS RULE ADM for more information about rules and rule sets



#### **Syntax**

Table 45-9 CREATE\_CAPTURE Procedure Parameters

Parameter	Description
queue_name	The name of the queue into which the capture process enqueues changes. You must specify an existing queue in the form [schema_name.] queue_name. For example, to specify a queue in the hr schema named streams_queue, enter hr.streams_queue. If the schema is not specified, then the current user is the default.
	<b>Note:</b> The queue_name setting cannot be altered after the capture process is created.
capture_name	The name of the capture process being created. A NULL specification is not allowed. Do not specify an owner.  Note: The capture_name setting cannot be altered after the capture process is created.
rule_set_name	The name of the positive rule set for the capture process. The positive rule set contains the rules that instruct the capture process to capture changes.
	If you want to use a positive rule set for the capture process, then you must specify an existing rule set in the form [schema_name.]rule_set_name. For example, to specify a positive rule set in the hr schema named job_capture_rules, enter hr.job_capture_rules. If the schema is not specified, then the current user is the default.
	An error is returned if the specified rule set does not exist.  You can create a rule set and add rules to it using the  DBMS RULE ADM package.
	If you specify NULL, and no negative rule set is specified, then the capture process captures all supported changes to all objects in the database, excluding database objects in the SYS and SYSTEM schemas.
	If you specify NULL, and a negative rule set exists for the capture process, then the capture process captures all changes that are not discarded by the negative rule set.

Table 45-9 (Cont.) CREATE\_CAPTURE Procedure Parameters

Parameter	Description
start_scn	A valid SCN for the database from which the capture process starts capturing changes.
	An error is returned if an invalid SCN is specified.
	The start_scn and start_time parameters are mutually exclusive.
	<b>See Also:</b> "Usage Notes" for more information setting the start_scn parameter
source_database	The global name of the source database. The source database is where the changes to be captured originated.
	If you do not include the domain name, then the procedure appends it to the database name automatically. For example, if you specify DBS1 and the domain is .EXAMPLE.COM, then the procedure specifies DBS1.EXAMPLE.COM automatically.
	If NULL, or if the specified name is the same as the global name of the current database, then local capture is assumed and only the default values for use_database_link and first_scn can be specified.
use_database_link	If TRUE, then the capture process at a downstream database uses a database link to the source database for administrative purposes relating to the capture process. A database link with the same name as the global name of the source database must exist at the downstream database.
	The capture process uses the database link to prepare database objects for instantiation at the source database and run the <code>DBMS_CAPTURE_ADM.BUILD</code> procedure at the source database, if necessary.
	During the creation of a downstream capture process, if the first_scn parameter is set to NULL, then the use_database_link parameter must be set to TRUE.  Otherwise, an error is returned.
	If FALSE, then either the capture process is running on the source database, or the capture process at a downstream database does not use a database link to the source database. In this case, you must perform the following administrative tasks manually:
	<ul> <li>Run the DBMS_CAPTURE_ADM.BUILD procedure at the source database to extract the data dictionary at the source database to the redo log when a capture process is created.</li> </ul>
	<ul> <li>Obtain the first SCN for the downstream capture process if the first SCN is not specified during capture process creation. The first SCN is needed to create and maintain a capture process.</li> </ul>
	Prepare source database objects for instantiation.



Table 45-9 (Cont.) CREATE\_CAPTURE Procedure Parameters

Parameter	Description
first_scn	The lowest SCN in the redo log from which a capture process can capture changes. A non-NULL value for this parameter is valid only if the DBMS_CAPTURE_ADM.BUILD procedure has been run at least once at the source database.
	You can query the FIRST_CHANGE# column of the V\$ARCHIVED_LOG dynamic performance view where the DICTIONARY_BEGIN column is YES to determine whether the DBMS_CAPTURE_ADM.BUILD procedure has been run on a source database. Any of the values returned by such a query can be used as a first_scn value if the redo log containing that SCN value is still available.
	<b>See Also:</b> "Usage Notes" for more information setting the first_scn parameter
logfile_assignment	If implicit, which is the default, then the capture process at a downstream database scans all redo log files added by redo transport services or manually from the source database to the downstream database.
	If explicit, then a redo log file is scanned by a capture process at a downstream database only if the capture process name is specified in the FOR <code>logminer_session_name</code> clause. If explicit, then the redo log file must be added manually to the downstream
	database, and redo transport services cannot be used to add redo log files to the capture process being created.
	If you specify explicit for this parameter for a local capture process, then the local capture process cannot use the online redo log to find changes. In this case, the capture process must use the archived redo log.
	<b>See Also:</b> "Usage Notes" for information about adding redo log files manually



Table 45-9 (Cont.) CREATE\_CAPTURE Procedure Parameters

### Parameter Description The name of the negative rule set for the capture process. negative rule set name The negative rule set contains the rules that instruct the capture process to discard changes. If you want to use a negative rule set for the capture process, then you must specify an existing rule set in the form [schema name.] rule set name. For example, to specify a negative rule set in the hr schema named neg capture rules, enter hr.neg capture rules. If the schema is not specified, then the current user is the default. If you specify NULL, and no positive rule set is specified, then the capture process captures all supported changes to all objects in the database, excluding database objects in the SYS and SYSTEM schemas. If you specify NULL, and a positive rule set exists for the capture process, then the capture process captures all changes that are not discarded by the positive rule set. An error is returned if the specified rule set does not exist. You can create a rule set and add rules to it using the DBMS RULE ADM package. If you specify both a positive and a negative rule set for a capture process, then the negative rule set is always evaluated first. The user in whose security domain a capture process capture user captures changes that satisfy its rule sets and runs custom rule-based transformations configured for capture process rules. If NULL, then the user who runs the CREATE CAPTURE procedure is used. Note: If the capture user for a capture process is dropped using DROP USER . . . CASCADE, then the capture process is also dropped automatically. See Also: "Usage Notes" for more information about this parameter. Either specify the number of days that a capture process checkpoint retention time retains checkpoints before purging them automatically, or specify DBMS CAPTURE ADM. INFINITE if checkpoints should not be purged automatically. If a number is specified, then a capture process purges a checkpoint the specified number of days after the checkpoint was taken. Partial days can be specified using decimal values. For example, .25 specifies 6 hours. When a checkpoint is purged, LogMiner data dictionary information for the archived redo log file that corresponds to the checkpoint is purged, and the first son of the capture process is reset to the SCN value corresponding to the first

change in the next archived redo log file.



Table 45-9 (Cont.) CREATE\_CAPTURE Procedure Parameters

Parameter	Description
start_time	A valid time from which the capture process starts capturing changes.
	An error is returned if an invalid time is specified.
	The start_scn and start_time parameters are mutually exclusive.
	<b>See Also:</b> "Usage Notes" for more information setting the start_time parameter
source root name	The global name of the root in the source CDB.
	If you specify $\mathtt{NULL}$ , or if the specified name is the same as the global name of the current root, then local capture is assumed.
	If not NULL, then remote capture is assumed and a condition is added to the generated rules to filter the LCRs based on the root in the source CDB.
	Note: In a downstream capture configuration, if the capture database is CDB and the source database is a non-CDB, then specify the same value for <code>source_root_name</code> and <code>source_database</code> . The <code>source_root_name</code> parameter does not need to be specified for local capture for either a CDB or a non-CDB.
capture_class	The valid values are 'Streams', 'XStream' or 'GoldenGate'.
	If NULL is specified, then 'Streams' is assumed.
	Note: The capture_class parameter cannot be set to 'Streams' or NULL when the capture database is a CDB.

Consider the following usage notes when you run this procedure:

- DBA Role Requirement
- Capture User Requirements
- First SCN and Start SCN Settings
- Explicit Log File Assignment

### **DBA Role Requirement**

If the user who invokes this procedure is different from the user specified in the <code>capture\_user</code> parameter, then the invoking user must be granted the <code>DBA</code> role. If the user who invokes this procedure is the same as the user specified in the <code>capture\_user</code> parameter, then the <code>DBA</code> role is not required for the invoking user. Only the <code>SYS</code> user can set the <code>capture\_user</code> user to <code>SYS</code>.

### Capture User Requirements

The <code>capture\_user</code> parameter specifies the user who captures changes that satisfy the capture process rule sets. This user must have the necessary privileges to capture changes. This procedure grants the capture user enqueue privilege on the queue used by the capture process and configures the user as a secure queue user of the queue.

In addition, ensure that the capture user has the following privileges:

- EXECUTE privilege on the rule sets used by the capture process
- EXECUTE privilege on all rule-based transformation functions used in the positive rule set

These privileges can be granted directly to the capture user, or they can be granted through roles.

In addition, the capture user must be granted EXECUTE privilege on all packages, including Oracle-supplied packages, that are invoked in rule-based transformations run by the capture process. These privileges must be granted directly to the capture user. They cannot be granted through roles.

### Note:

- A capture user does not require privileges on a database object to capture changes to the database object. The capture process can pass these changes to a rule-based transformation function. Therefore, ensure that you consider security implications when you configure a capture process.
- Creation of the first capture process in a database might take some time because the data dictionary is duplicated during this creation.

#### First SCN and Start SCN Settings

When you create a capture process using this procedure, you can specify the first SCN and start SCN for the capture process. A capture process scans the redo data from the first SCN or an existing capture process checkpoint forward, even if the start SCN is higher than the first SCN or the checkpoint SCN. In this case, the capture process does not capture any changes in the redo data before the start SCN. Oracle recommends that, at capture process creation time, the difference between the first SCN and start SCN be as small as possible to keep the amount of redo scanned by the capture process to a minimum.

#### Note:

When you specify the <code>start\_time</code> parameter instead of the <code>start\_scn</code> parameter, the <code>start\_time</code> corresponds with a specific SCN. In this case, the information in this section also applies to the SCN that corresponds with the specified <code>start\_time</code>.

In some cases, the behavior of the capture process is different depending on the settings of these SCN values and on whether the capture process is local or downstream.

The following table describes capture process behavior for SCN value settings:



first_scn Setting	start_scn Setting	Capture Process Type	Description
Non-NULL NULL	Local or Downstream	The new capture process is created at the local database with a new LogMiner session starting from the value specified for the first_scn parameter. The start SCN is set to the specified first SCN value automatically, and the new capture process does not capture changes that were made before this SCN.	
			The BUILD procedure in the DBMS_CAPTURE_ADM package is not run automatically. This procedure must have been run at least once before on the source database, and the specified first SCN must correspond to the SCN value of a previous build that is still available in the redo log. When the new capture process is started for the first time, it creates a new LogMiner data dictionary using the data dictionary information in the redo log. If the BUILD procedure has not been run at least once on the source database, then an error is raised when the capture process is started.
		Capture process behavior is the same for a local capture process and a downstream capture process created with these SCN settings, except that a local capture process is created at the source database and a downstream capture process is created at the downstream database.	
Non-NULL	Non-NULL	Local or Downstream	If the specified value for the <code>start_scn</code> parameter is greater than or equal to the specified value for the <code>first_scn</code> parameter, then the new capture process is created at the local database with a new LogMiner session starting from the specified first SCN. In this case, the new capture process does not capture changes that were made before the specified start SCN. If the specified value for the <code>start_scn</code> parameter is less than the specified value for the <code>first_scn</code> parameter, then an error is raised.
			The BUILD procedure in the DBMS_CAPTURE_ADM package is not run automatically. This procedure must have been called at least once before on the source database, and the specified first_scn must correspond to the SCN value of a previous build that is still available in the redo log. When the new capture process is started for the first time, it creates a new LogMiner data dictionary using the data dictionary information in the redo log. If the BUILD procedure has not been run at least once on the source database, then an error is raised.
			Capture process behavior is the same for a local capture process and a downstream capture process created with these SCN settings, except that a local capture process is created at the source database and a downstream capture process is created at the downstream database.



first_scn Setting	start_scn Setting	Capture Process Type	Description
NULL	Non-NULL	Local	The new capture process creates a new LogMiner data dictionary if either one of the following conditions is true:
			<ul> <li>There is no existing capture process for the local source database, and the specified value for the start_scn parameter is greater than or equal to the current SCN for the database.</li> </ul>
			<ul> <li>There are existing capture processes, but none of the capture processes have taken a checkpoint yet, and the specified value for the start_scn parameter is greater than or equal to the current SCN for the database.</li> </ul>
		In either of these cases, the <code>BUILD</code> procedure in the <code>DBMS_CAPTURE_ADM</code> package is run during capture process creation. The new capture process uses the resulting build of the source data dictionary in the redo log to create a LogMiner data dictionary the first time it is started, and the first SCN corresponds to the SCN of the data dictionary build. If there are any in-flight transactions, then the <code>BUILD</code> procedure waits until these transactions commit before completing. An in-flight transaction is one that is active during capture process creation or a data dictionary build.	
	source database that has to process shares an existing the existing capture procest SCN that is lower than or estarted successfully at least transactions, then the captor	However, if there is at least one existing local capture process for the local source database that has taken a checkpoint, then the new capture process shares an existing LogMiner data dictionary with one or more of the existing capture processes. In this case, a capture process with a first SCN that is lower than or equal to the specified start SCN must have been started successfully at least once. Also, if there are any in-flight transactions, then the capture process is created after these transactions commit.	
		If there is no existing capture process for the local source database (or if no existing capture processes have taken a checkpoint yet), and the specified start SCN is less than the current SCN for the database, then an error is raised.	



first_scn Setting	start_scn Setting	Capture Process Type	Description
NULL Non-NULL De	Downstream	When the CREATE_CAPTURE procedure creates a downstream capture process, the use_database_link parameter must be set to TRUE when the first_scn parameter is set to NULL. Otherwise, an error is raised. The database link is used to obtain the current SCN of the source database.	
			The new capture process creates a new LogMiner data dictionary if either one of the following conditions is true:
			<ul> <li>There is no existing capture process that captures changes to the source database at the downstream database, and the specified value for the start_scn parameter is greater than or equal to the current SCN for the source database.</li> </ul>
		<ul> <li>There are existing capture processes that capture changes to the source database at the downstream database, but none of the capture processes have taken a checkpoint yet, and the specified value for the start_scn parameter is greater than or equal to the current SCN for the source database.</li> </ul>	
		In either of these cases, the <code>BUILD</code> procedure in the <code>DBMS_CAPTURE_ADM</code> package is run during capture process creation. The first time you start the new capture process, it uses the resulting build of the source data dictionary in the redo log files copied to the downstream database to create a LogMiner data dictionary. Here, the first SCN for the new capture process corresponds to the SCN of the data dictionary build. If there are any inflight transactions, then the <code>BUILD</code> procedure waits until these transactions commit before completing.	
		However, if at least one existing capture process has taken a checkpoint and captures changes to the source database at the downstream database, then the new capture process shares an existing LogMiner data dictionary with one or more of these existing capture processes. In this case, one of these existing capture processes with a first SCN that is lower than or equal to the specified start SCN must have been started successfully at least once. Also, if there are any in-flight transactions, then the capture process is created after these transactions commit.	
		If there is no existing capture process that captures changes to the source database at the downstream database (or no existing capture process has taken a checkpoint), and the specified <pre>start_scn</pre> parameter value is less than the current SCN for the source database, then an error is raised.	
NULL	NULL	Local or Downstream	The behavior is the same as setting the first_scn parameter to <code>NULL</code> and setting the <code>start_scn</code> parameter to the current SCN of the source database.

## See Also:

BUILD Procedure for more information about the  ${\tt BUILD}$  procedure and the LogMiner data dictionary

## Explicit Log File Assignment

If you specify explicit for the  $logfile\_assignment$  parameter, then you add a redo log file manually to a downstream database using the following statement:

```
ALTER DATABASE REGISTER LOGICAL LOGFILE file name FOR capture process;
```

Here, <code>file\_name</code> is the name of the redo log file being added and <code>capture\_process</code> is the name of the capture process that will use the redo log file at the downstream database. The <code>capture\_process</code> is equivalent to the <code>logminer\_session\_name</code> and must be specified. The redo log file must be present at the site running the downstream database. You must transfer this file manually to the site running the downstream database using the <code>DBMS\_FILE\_TRANSFER</code> package, FTP, or some other transfer method.



Oracle Database SQL Language Reference for more information about the ALTER DATABASE statement and Oracle Data Guard Concepts and Administration for more information registering redo log files

# CREATE\_SYNC\_CAPTURE Procedure

This procedure creates a synchronous capture.

#### **Syntax**

Table 45-10 CREATE\_SYNC\_CAPTURE Procedure Parameters

Parameter	Description
——————————————————————————————————————	Description
queue_name	The name of the queue into which the synchronous capture enqueues changes. You must specify an existing queue in the form [schema_name.] queue_name. For example, to specify a queue in the strmadmin schema named streams_queue, enter strmadmin.streams_queue. If the schema is not specified, then the current user is the default.
	•
	<b>Note:</b> The queue_name setting cannot be altered after the synchronous capture is created.
capture_name	The name of the synchronous capture being created. A ${\tt NULL}$ specification is not allowed. Do not specify an owner.
	<b>Note:</b> The capture_name setting cannot be altered after the synchronous capture is created.
rule_set_name	The name of the positive rule set for the synchronous capture. The positive rule set contains the rules that instruct the synchronous capture to capture changes.
	Specify an existing rule set in the form [schema_name.] rule_set_name. For example, to specify a positive rule set in the strmadmin schema named sync_cap_rules, enter strmadmin.sync_cap_rules. If the schema is not specified, then the current user is the default.  An error is returned if the specified rule set does not exist.  If NULL, then an error is returned.

Table 45-10 (Cont.) CREATE\_SYNC\_CAPTURE Procedure Parameters

Parameter	Description
capture_user	The user in whose security domain the synchronous capture captures changes that satisfy its rule set and runs custom rule-based transformations configured for synchronous capture rules. If NULL, then the user who runs the CREATE_SYNC_CAPTURE procedure is used.
	Only a user who is granted the DBA role can set a capture user. Only the SYS user can set the capture_user to SYS.
	<b>Note:</b> If the capture user for a synchronous capture is dropped using DROP USER CASCADE, then the synchronous capture is also dropped automatically.
	See Also: "Usage Notes" for more information about this parameter.

When the CREATE\_SYNC\_CAPTURE procedure creates a synchronous capture, the procedure must obtain an exclusive lock on each table for which it will capture changes. The rules in the specified rule set for the synchronous capture determine these tables. If there are outstanding transactions on a table for which the synchronous capture will capture changes, then the procedure waits until it can obtain a lock.

The capture\_user parameter specifies the user who captures changes that satisfy the synchronous capture rule set. This user must have the necessary privileges to capture changes.

In addition, ensure that the capture user has the following privileges:

- ENQUEUE privilege on the queue specified in the queue name parameter
- EXECUTE privilege on the rule set used by the synchronous capture
- EXECUTE privilege on all rule-based transformation functions used in the rule set

These privileges can be granted directly to the capture user, or they can be granted through roles.

In addition, the capture user must be granted EXECUTE privilege on all packages, including Oracle-supplied packages, that are invoked in rule-based transformations run by the synchronous capture. These privileges must be granted directly to the capture user. These privileges cannot be granted through roles.



A capture user does not require privileges on a database object to capture changes to the database object. The synchronous capture can pass these changes to a rule-based transformation function. Therefore, ensure that you consider security implications when you configure a synchronous capture.



# DROP\_CAPTURE Procedure

This procedure drops a capture process.

#### **Syntax**

#### **Parameters**

#### Table 45-11 DROP\_CAPTURE Procedure Parameters

Parameter	Description
capture_name	The name of the capture process being dropped. Specify an existing capture process name. Do not specify an owner.
drop_unused_rule_sets	If TRUE, then the procedure drops any rule sets, positive and negative, used by the specified capture process if these rule sets are not used by any other Oracle Replication client. Oracle Replication clients include capture processes, propagations, apply processes, and messaging clients. If this procedure drops a rule set, then this procedure also drops any rules in the rule set that are not in another rule set.  If FALSE, then the procedure does not drop the rule sets used by the specified capture process, and the rule sets retain their rules.

#### **Usage Notes**

The following usage notes apply to this procedure:

- The Capture Process Must Be Stopped Before It Is Dropped
- The DROP\_CAPTURE Procedure and Rules-related Information

The Capture Process Must Be Stopped Before It Is Dropped

A capture process must be stopped before it can be dropped.

```
See Also:

STOP_CAPTURE Procedure
```

The DROP\_CAPTURE Procedure and Rules-related Information

When you use this procedure to drop a capture process, rules-related information for the capture process is removed from the data dictionary views for Oracle Replication rules. Information about such a rule is removed even if the rule is not in either rule set for the capture process.

The following are the data dictionary views for Oracle Replication rules:

- ALL STREAMS GLOBAL RULES
- DBA STREAMS GLOBAL RULES



- ALL\_STREAMS\_SCHEMA\_RULES
- DBA\_STREAMS\_SCHEMA\_RULES
- ALL\_STREAMS\_TABLE\_RULES
- DBA\_STREAMS\_TABLE\_RULES
- DBA STREAMS RULES

# INCLUDE\_EXTRA\_ATTRIBUTE Procedure

This procedure includes or excludes an extra attribute in logical change records (LCRs) captured by the specified capture process or synchronous capture.

### **Syntax**

Table 45-12 INCLUDE\_EXTRA\_ATTRIBUTE Procedure Parameters

Parameter	Description		
capture_name	The name of the capture process or synchronous capture. Specify an existing capture process name or synchronous capture name. Do not specify an owner.		
attribute_name	The name of the attribute to be included in or excluded from LCRs captured by the capture process or synchronous capture. The following names are valid settings:		
	<ul><li>row_id</li></ul>		
	The rowid of the row changed in a row LCR. This attribute is not included in DDL LCRs, or in row LCRs for index-organized tables. The type is VARCHAR2.		
	• serial#		
	The serial number of the session that performed the change captured in the LCR. The type is <code>NUMBER</code> .		
	• session#		
	The identifier of the session that performed the change captured in the LCR. The type is NUMBER.		
	• thread#		
	The thread number of the instance in which the change captured in the LCR was performed. Typically, the thread number is relevant only in an Oracle Real Application Clusters (Oracle RAC) environment. The type is NUMBER.		
	• tx_name		
	The name of the transaction that includes the LCR. The type is VARCHAR2.		
	• username		
	The name of the user who performed the change captured in the LCR. The type is VARCHAR2.		
include	If TRUE, then the specified attribute is included in LCRs captured by the capture process or synchronous capture.		
	If FALSE, then the specified attribute is excluded from LCRs captured by the capture process or synchronous capture.		

Some information is not captured by a capture process or synchronous capture unless you use this procedure to specify that the information should be captured. If you want to exclude an extra attribute that is being captured by a capture process or synchronous capture, then specify the attribute and specify FALSE for the include parameter.

# PREPARE\_GLOBAL\_INSTANTIATION Procedure

This procedure performs the synchronization necessary for instantiating all the tables in the database at another database and can enable supplemental logging for key columns or all columns in these tables.

This procedure prepares the tables in the database for instantiation when a capture process will be used to capture changes to the tables in the database.

This procedure records the lowest SCN of each object in the database for instantiation. SCNs after the lowest SCN for an object can be used for instantiating the object. Running this procedure prepares all current and future objects in the database for instantiation.

### **Syntax**

#### **Parameter**

#### Table 45-13 PREPARE\_GLOBAL\_INSTANTIATION Procedure Parameter

### Parameter Description supplemental logging Either NONE, KEYS, or ALL. If NONE is specified, then this procedure does not enable supplemental logging for any columns in the tables in the database. This procedure does not remove existing supplemental logging specifications for these tables. If KEYS is specified, then this procedure enables supplemental logging for primary key, unique key, bitmap index, and foreign key columns in the tables in the database and for any table added to the database in the future. Primary key columns are logged unconditionally. Unique key, bitmap index, and foreign key columns are logged conditionally. Specifying KEYS does not enable supplemental logging of bitmap join index columns. If ALL is specified, then this procedure enables supplemental logging for all columns in the tables in the database and for any table added to the database in the future. The columns are logged unconditionally. Supplemental logging is not enabled for columns of the following types: LOB, LONG, LONG RAW, user-defined types, and Oracle-supplied types.



Table 45-13 (Cont.) PREPARE\_GLOBAL\_INSTANTIATION Procedure Parameter

Parameter	Description
container	Either CURRENT, ALL, or pdb_name.
	If CURRENT is specified, then this procedure adds supplemental logging for the current container.
	If $\mathtt{ALL}$ is specified, then this procedure adds supplemental logging for all of the containers in the current CDB.
	If pdb_name is specified, then this procedure adds supplemental logging for the specified PDB.
	$\mathtt{ALL}$ and $pdb\_name$ are valid only when you invoke the procedure from the root.

Run this procedure at the source database.

If you use a capture process to capture all of the changes to a database, then use this procedure to prepare the tables in the database for instantiation after the capture process has been configured.

# PREPARE\_SCHEMA\_INSTANTIATION Procedure

This procedure performs the synchronization necessary for instantiating all tables in the schema at another database and can enable supplemental logging for key columns or all columns in these tables.

This procedure prepares the tables in the schema for instantiation when a capture process will be used to capture changes to the tables in the schema.

This procedure records the lowest SCN of each object in the schema for instantiation. SCNs after the lowest SCN for an object can be used for instantiating the object. Running this procedure prepares all current and future objects in the schema for instantiation.

#### **Syntax**

```
DBMS_CAPTURE_ADM.PREPARE_SCHEMA_INSTANTIATION(
schema_name IN VARCHAR2,
supplemental_logging IN VARCHAR2 DEFAULT 'KEYS',
container IN VARCHAR2 DEFAULT 'CURRENT');
```

Table 45-14 PREPARE\_SCHEMA\_INSTANTIATION Procedure Parameters

Parameter	Description	
schema name	The name of the schema. For example, hr.	



Table 45-14 (Cont.) PREPARE\_SCHEMA\_INSTANTIATION Procedure Parameters

Parameter	Description		
supplemental_logging	Either NONE, KEYS, ALL, PARTIAL_JSON_OFF, or PARTIAL_JSON_ON.		
	If NONE is specified, then this procedure does not enable supplemental logging for any columns in the tables in the schema. This procedure does not remove existing supplemental logging specifications for these tables.		
	If KEYS is specified, then this procedure enables supplemental logging for primary key, unique key, bitmap index, and foreign key columns in the tables in the schema and for any table added to this schema in the future. Primary key columns are logged unconditionally. Unique key, bitmap index, and foreign key columns are logged conditionally. Specifying KEYS does not enable supplemental logging of bitmap join index columns.		
	If ALL is specified, then this procedure enables supplemental logging for all columns in the tables in the schema and for any table added to this schema in the future. The columns are logged unconditionally. Supplemental logging is not enabled for columns of the following types: LOB, LONG, LONG RAW, user-defined types, and Oracle-supplied types.		
	If PARTIAL_JSON_OFF is specified, partial JSON update is disabled at schema level. If PARTIAL_JSON_ON is specified, partial JSON update is enabled at schema level. If PARTIAL_JSON_ON or PARTIAL_JSON_OFF is unspecified, schema level partial JSON setting will be unchanged. If PARTIAL_JSON_OFF has never been specified, by default, partial JSON update is disabled at schema level.		
container	Either CURRENT, ALL, or pdb_name.		
	If CURRENT is specified, then this procedure adds supplemental logging for the current container.		
	If ${\tt ALL}$ is specified, then this procedure adds supplemental logging for all of the containers in the current CDB.		
	If pdb_name is specified, then this procedure adds supplemental logging for the specified PDB		
	ALL and $pdb\_name$ are valid only when you invoke the procedure from the root.		

Run this procedure at the source database. If you use a capture process to capture all of the changes to a schema, then use this procedure to prepare the tables in the schema for instantiation after the capture process has been configured.

# PREPARE\_SYNC\_INSTANTIATION Function

This function performs the synchronization necessary for instantiating one or more tables at another database. This function returns the prepare system change number (SCN) for the table or tables being prepared for instantiation.

This function prepares one or more tables for instantiation when a synchronous capture will be used to capture changes to the tables.

This function records the lowest SCN of each table for instantiation (prepare SCN). SCNs after the lowest SCN for an object can be used for instantiating the object.

This function is overloaded. The table\_names parameter is VARCHAR2 datatype in one version and DBMS\_UTILITY.UNCL\_ARRAY datatype in the other version.

#### **Syntax**

```
DBMS_CAPTURE_ADM.PREPARE_SYNC_INSTANTIATION(
   table_names IN VARCHAR2)
RETURN NUMBER;

DBMS_CAPTURE_ADM.PREPARE_SYNC_INSTANTIATION(
   table_names IN DBMS_UTILITY.UNCL_ARRAY)
RETURN NUMBER;
```

#### **Parameters**

Table 45-15 PREPARE\_SYNC\_INSTANTIATION Function Parameter

Parameter	Description
table_names	When the table_names parameter is VARCHAR2 datatype, a comma-delimited list of the tables to prepare for instantiation. There must be no spaces between entries.
	When the table_names parameter is DBMS_UTILITY.UNCL_ARRAY datatype, specify a PL/SQL associative array of this type that contains the names of the tables to prepare for instantiation. The first table name is at position 1, the second at position 2, and so on. The table does not need to be NULL terminated.
	In either version of the function, specify the name of each table in the form [schema_name.]table_name. For example, hr.employees. If the schema is not specified, then the current user is the default.

# PREPARE\_TABLE\_INSTANTIATION Procedure

This procedure performs the synchronization necessary for instantiating the table at another database and can enable supplemental logging for key columns or all columns in the table

This procedure prepares the table for instantiation when a capture process will be used to capture changes to the table.

This procedure records the lowest SCN of the table for instantiation. SCNs after the lowest SCN for an object can be used for instantiating the object.

### **Syntax**

```
DBMS_CAPTURE_ADM.PREPARE_TABLE_INSTANTIATION(
table_name IN VARCHAR2,
supplemental_logging IN VARCHAR2 DEFAULT 'KEYS',
container IN VARCHAR2 DEFAULT 'CURRENT');
```

Table 45-16 PREPARE\_TABLE\_INSTANTIATION Procedure Parameters

Parameter	Description
table_name	The name of the table specified as [schema_name.] object_name. For
	example, hr.employees. If the schema is not specified, then the current
	user is the default.



Table 45-16 (Cont.) PREPARE\_TABLE\_INSTANTIATION Procedure Parameters

Parameter	Description		
supplemental_logging	Either NONE, KEYS, or ALL.		
	If NONE is specified, then this procedure does not enable supplemental logging for any columns in the table. This procedure does not remove existing supplemental logging specifications for the table.		
	If KEYS is specified, then this procedure enables supplemental logging for primary key, unique key, bitmap index, and foreign key columns in the table. The procedure places the key columns for the table in three separate log groups: the primary key columns in an unconditional log group, the unique key columns and bitmap index columns in a conditional log group, and the foreign key columns in a conditional log group. Specifying KEYS does not enable supplemental logging of bitmap join index columns.		
	If ALL is specified, then this procedure enables supplemental logging for all columns in the table. The procedure places all of the columns for the table in an unconditional log group. Supplemental logging is not enabled for columns of the following types: LOB, LONG, LONG RAW, user-defined types, and Oracle-supplied types.		
container	Either CURRENT, ALL, or pdb_name.		
	If CURRENT is specified, then this procedure adds supplemental logging for the current container.		
	If ${\tt ALL}$ is specified, then this procedure adds supplemental logging for all of the containers in the current CDB.		
	If pdb_name is specified, then this procedure adds supplemental logging for the specified PDB.		
	${\tt ALL}$ and ${\tt pdb\_name}$ are valid only when you invoke the procedure from the root.		

Run this procedure at the source database. If you use a capture process to capture all of the changes to a table, then use this procedure to prepare the table for instantiation after the capture process has been configured.

# SET\_PARAMETER Procedure

This procedure sets a capture process parameter to the specified value.

### **Syntax**

```
DBMS_CAPTURE_ADM.SET_PARAMETER(
capture_name IN VARCHAR2,
parameter IN VARCHAR2,
value IN VARCHAR2 DEFAULT NULL);
```

Table 45-17 SET PARAMETER Procedure Parameters

Parameter	Description
capture_name	The name of the capture process. Do not specify an owner.

Table 45-17 (Cont.) SET\_PARAMETER Procedure Parameters

Parameter	Description	
parameter	The name of the parameter you are setting.	
value	The value to which the parameter is set.  If NULL, then the parameter is set to its default value.	

## **Capture Process Parameters**

The following table lists the parameters for the capture process.

**Table 45-18 Capture Process Parameters** 

Parameter Name	Possible Values	Default	Description
annotateddlsize	0 or a positive integer up to 4294967296 (4 GB)	0	The maximum length of the DDL text, in bytes, up to which the DDL can be chosen for annotation, if applicable for the DDL. Any DDL text whose size is greater than the specified size is not annotated.
			The default, 0, indicates that no DDL text processed by this capture process is annotated.
			Annotating DDL can simplify filtering and transformation of the DDL within Oracle GoldenGate.
			<b>Note:</b> This parameter is intended for Oracle GoldenGate. Do not use this parameter in an Oracle Replication environment or in an XStream environment.
<pre>capture_idkey_obje cts</pre>	Y Or N	N	If Y, then the capture process captures ID key logical change records (LCRs).
			If $\mathbb{N}$ , then the capture process does not capture ID key LCRs.
			Capture processes do not fully support capturing changes to some data types from the redo log. ID key LCRs enable an XStream configuration to capture these changes and process them with an XStream client application.
			Note: This parameter is intended for XStream.  Do not use this parameter in an Oracle Replication environment unless XStream optimizations are enabled by the  DBMS_XSTREAM_ADM.ENABLE_GG_XSTREAM_FO R_STREAMS procedure. See "ENABLE_GG_XSTREAM_FOR_STREAMS Procedure" for information about enabling XStream optimizations.
			<b>See Also:</b> "Usage Notes" for more information about this parameter and <i>Oracle Database XStream Guide</i> for more information about ID key LCRs



Table 45-18 (Cont.) Capture Process Parameters

Parameter Name	Possible Values	Default	Description
capture_sequence_n extval	Y or N	N	If Y, then the capture process captures sequence LCRs for all of the sequences in the database, except for sequences in Oraclesupplied, administrative schemas such as SYS and SYSTEM. The capture process's rule sets can filter sequence LCRs in the same way that they filter row LCRs and DDL LCRs.
			If $\mathbb{N}$ , then the capture process does not capture sequence LCRs.
			An apply process or XStream inbound server can use sequence LCRs to ensure that the sequence values at a destination database use the appropriate values. For increasing sequences, the sequence values at the destination are equal to or greater than the sequence values at the source database. For decreasing sequences, the sequence values at the destination are less than or equal to the sequence values at the source database.
			Note: This parameter is intended for XStream.  Do not use this parameter in an Oracle Replication environment unless XStream optimizations are enabled by the  DBMS_XSTREAM_ADM.ENABLE_GG_XSTREAM_FO R_STREAMS procedure. See "ENABLE_GG_XSTREAM_FOR_STREAMS Procedure" for information about enabling XStream optimizations.
			See Also: "Usage Notes" for more information about this parameter and "SET_PARAMETER Procedure" for information about the apply_sequence_nextval apply process parameter
disable_on_limit	Y or N	N	If Y, then the capture process is disabled because it reached a value specified by the time_limit parameter or message_limit parameter.
			If $\mathbb{N},$ then the capture process is restarted immediately after stopping because it reached a limit.
			When a capture process is restarted, it starts to capture changes at the point where it last stopped. A restarted capture process gets a new session identifier, and the processes associated with the capture process also get new session identifiers. However, the capture process number (CPnn) remains the same.



Table 45-18 (Cont.) Capture Process Parameters

Parameter Name	Possible Values	Default	Description
downstream_real_ti me_mine	Y or N	Y for local capture processes N for downstream capture processes	If Y, then the capture process is a real-time downstream capture process. After setting this parameter to Y, switch the redo log file at the source database using the SQL statement ALTER SYSTEM ARCHIVE LOG CURRENT to begin real-time downstream capture. If this parameter is set to Y, then redo data from the source database must be sent to the standby redo log at the downstream database.  If N, then the capture process is an archived-log downstream capture process.  An error is raised if an attempt is made to set this parameter for a local capture process.
excludetag	Comma-delimited list of Oracle Replication tags	NULL	Controls whether the capture process captures DML changes that are tagged with one of the specified Oracle Replication tags.
			Whether the capture process captures these changes depends on the settings for the getapplops and getreplicates parameters.
			If ${\tt NULL},$ then the capture process ignores this
			parameter.
			Note: This parameter is intended for XStream. Do not use this parameter in an Oracle Replication environment unless XStream optimizations are enabled by the DBMS_XSTREAM_ADM.ENABLE_GG_XSTREAM_FOR_STREAMS procedure. See "ENABLE_GG_XSTREAM_FOR_STREAMS Procedure" for information about enabling XStream optimizations.
			<b>See Also:</b> "Usage Notes" for more information about this parameter
excludetrans	Comma-delimited list of transaction names	NULL	Controls whether the capture process captures DML changes in the specified transaction names.
			Whether the capture process captures these changes depends on the settings for the getapplops and getreplicates parameters.
			If $\mathtt{NULL},$ then the capture process ignores this parameter.
			Note: This parameter is intended for XStream. Do not use this parameter in an Oracle Replication environment unless XStream optimizations are enabled by the DBMS_XSTREAM_ADM.ENABLE_GG_XSTREAM_FOR_STREAMS procedure. See "ENABLE_GG_XSTREAM_FOR_STREAMS Procedure" for information about enabling XStream optimizations.
			See Also: "Usage Notes" for more information about this parameter



Table 45-18 (Cont.) Capture Process Parameters

Parameter Name	Possible Values	Default	Description
excludeuser	Comma-delimited list of user names	NULL	Controls whether the capture process captures DML changes made by the specified users.
			Whether the capture process captures these changes depends on the settings for the getapplops and getreplicates parameters.
			Specify an exact pattern match for each user name. The pattern match is case sensitive. For example, specify HR for the hr user.
			If NULL, then the capture process ignores this parameter.
			Note: This parameter is intended for XStream.  Do not use this parameter in an Oracle Replication environment unless XStream optimizations are enabled by the  DBMS_XSTREAM_ADM.ENABLE_GG_XSTREAM_FO R_STREAMS procedure. See "ENABLE_GG_XSTREAM_FOR_STREAMS Procedure" for information about enabling XStream optimizations.
			<b>See Also:</b> "Usage Notes" for more information about this parameter
excludeuserid	Comma-delimited list of user ID values	NULL	Controls whether the capture process captures data manipulation language (DML) changes made by the specified users.
			Whether the capture process captures these changes depends on the settings for the getapplops and getreplicates parameters.
			To view the user ID for a user, query the USER_ID column in the ALL_USERS data dictionary view.
			If NULL, then the capture process ignores this parameter.
			Note: This parameter is intended for XStream.  Do not use this parameter in an Oracle Replication environment unless XStream optimizations are enabled by the  DBMS_XSTREAM_ADM.ENABLE_GG_XSTREAM_FO R_STREAMS procedure. See "ENABLE_GG_XSTREAM_FOR_STREAMS Procedure" for information about enabling XStream optimizations.
			<b>See Also:</b> "Usage Notes" for more information about this parameter



Table 45-18 (Cont.) Capture Process Parameters

Parameter Name	Possible Values	Default	Description
getapplops	Y or N	Y	If Y, then the capture process captures DML changes if the original user is not specified in the excludeuserid or excludeuser parameters and the transaction name is not specified in the excludetrans parameter.
			If N, then the capture process ignores DML changes if the original user is not specified in the excludeuserid or excludeuser parameters and the transaction name is not specified in the excludetrans parameter.
			In either case, the capture process captures a DML change only if it satisfies the capture process's rule sets.
			Note: This parameter is intended for XStream.  Do not use this parameter in an Oracle Replication environment unless XStream optimizations are enabled by the  DBMS_XSTREAM_ADM.ENABLE_GG_XSTREAM_FO R_STREAMS procedure. See "ENABLE_GG_XSTREAM_FOR_STREAMS Procedure" for information about enabling XStream optimizations.
			<b>See Also:</b> "Usage Notes" for more information about this parameter
getreplicates	Y or N	N	If Y, then the capture process captures DML changes if the original user is specified in the excludeuserid or excludeuser parameters and the transaction name is specified in the excludetrans parameter.
			If N, then the capture process ignores DML changes if the original user is specified in the excludeuserid or excludeuser parameters and the transaction name is specified in the excludetrans parameter.
			In either case, the capture process captures a DML change only if it satisfies the capture process's rule sets.
			Note: This parameter is intended for XStream.  Do not use this parameter in an Oracle Replication environment unless XStream optimizations are enabled by the  DBMS_XSTREAM_ADM.ENABLE_GG_XSTREAM_FO R_STREAMS procedure. See "ENABLE_GG_XSTREAM_FOR_STREAMS Procedure" for information about enabling XStream optimizations.
			<b>See Also:</b> "Usage Notes" for more information about this parameter



Table 45-18 (Cont.) Capture Process Parameters

Parameter Name	Possible Values	Default	Description
<pre>ignore_transaction</pre>	A valid transaction ID or NULL	NULL	Instructs the capture process to ignore the specified transaction from the source database, effective immediately.
			The capture process eliminates all subsequent LCRs for the transaction. If the specified transaction is committed successfully at the source database, the destination database will receive a ROLLBACK statement instead, and any LCRs from the transaction that were enqueued before the ignore transaction request are rolled backed at the destination database.
			If NULL, then the capture process ignores this parameter.
			Use caution when setting this parameter because ignoring a transaction might lead to data divergence between the source database and destination database.
			To ignore multiple transactions, specify each transaction in a separate call to the SET_PARAMETER procedure. The DBA_CAPTURE_PARAMETERS view displays a comma-delimited list of all transactions to be ignored. To clear the list of ignored transactions, run the SET_PARAMETER procedure and specify NULL for the ignore_transaction parameter.
	A fully qualified table name, *, or -	*	Controls the behavior of the capture process when it tries to capture changes to a specified table or to an unsupported table.
			A capture process tries to capture changes to an unsupported table when its rule sets instruct it to do so. If you do not want the capture process to try to capture changes to unsupported tables, then ensure that the capture process's rule sets exclude unsupported tables.
			When a table name is specified, the capture process does not capture changes to the specified table. The table name must be entered in the form <code>table_owner.table_name</code> . For example. hr.employees. To specify multiple tables, specify each table in a separate call to the <code>SET_PARAMETER</code> procedure.
			When * is specified and the capture process tries to capture a change to an unsupported table, the capture process ignores the change and continues to run. The change to the unsupported table is not captured, and the capture process records the unsupported table in the alert log.
			When – is specified and the capture process tries to capture a change to an unsupported table, the capture process aborts.



Table 45-18 (Cont.) Capture Process Parameters

Parameter Name	Possible Values	Default	Description
include_objects	A list of tables or schema names separated by commas	none	Directs capture to include changes from the specified tables or schemas.
			An LCR that is selected by include_objects is passed to the outbound server regardless of any further filtering that is specified.
			Note: This parameter is intended for XStream. Do not use or attempt to set this parameter in an Oracle Replication environment unless XStream optimizations are enabled by the DBMS_XSTREAM_ADM.ENABLE_GG_XSTREAM_FO R_STREAMS procedure. See "ENABLE_GG_XSTREAM_FOR_STREAMS Procedure" for information about enabling
			XStream optimizations.  See Also: "Usage Notes" for more information about this parameter
<pre>inline_lob_optimiz ation</pre>	Y or N	N	If Y, then LOBs that can be processed inline (such as small LOBs) are included in the LCR directly, rather than sending LOB chunk LCRs.
			If $\mathbb{N},$ then each LOB column is sent as $\mathbb{NULL}$ followed by LOB chunk LCRs to update the LOB column.
			Note: This parameter is intended for XStream. Do not use or attempt to set this parameter in an Oracle Replication environment unless XStream optimizations are enabled by the DBMS_XSTREAM_ADM.ENABLE_GG_XSTREAM_FO R_STREAMS procedure. See "ENABLE_GG_XSTREAM_FOR_STREAMS Procedure" for information about enabling XStream optimizations.
			<b>See Also:</b> "Usage Notes" for more information about this parameter
maximum_scn	A valid SCN or INFINITE	INFINITE	The capture process is disabled before capturing a change record with an SCN greater than or equal to the value specified.
			If INFINITE, then the capture process runs regardless of the SCN value.



Table 45-18 (Cont.) Capture Process Parameters

Parameter Name	Possible Values	Default	Description
max_sga_size	A positive integer	INFINITE	Controls the amount of system global area (SGA) memory allocated specifically to the capture process, in megabytes. The capture process attempts to allocate memory up to this limit. A capture process uses Oracle LogMiner to scan for changes in the redo log.
			The memory is allocated for the duration of the capture process session and is released when the capture process becomes disabled.
			<b>Note:</b> The sum of system global area (SGA) memory allocated for all components on a database must be less than the value set for the STREAMS_POOL_SIZE initialization parameter.
			If NULL, then the capture component uses the original default value. A NULL value has the same effect as resetting the parameter to its default value.
			Note: This parameter is intended for XStream. Do not use or attempt to set this parameter in an Oracle Replication environment unless XStream optimizations are enabled by the DBMS_XSTREAM_ADM.ENABLE_GG_XSTREAM_FO R_STREAMS procedure. See "ENABLE_GG_XSTREAM_FOR_STREAMS Procedure" for information about enabling XStream optimizations.
			See Also: "Usage Notes" for more information about this parameter
merge_threshold	A negative integer, 0, a positive integer, or INFINITE	60	The amount of time, in seconds, between the message creation time of the original capture process and the message creation time of the cloned capture process.
			Specifically, if the difference, in seconds, between the CAPTURE_MESSAGE_CREATE_TIME of the cloned capture process and the original capture process is less than or equal to the value specified for this parameter, then automatic merge begins by running the MERGE_STREAMS procedure.
			This parameter is relevant only when changes captured by the capture process are applied by two or more apply processes and the split_threshold parameter is set to a value other than INFINITE.
			If a negative value is specified, then automatic merge is disabled.
			If 0 (zero) is specified, then there must be no lag between the original capture process and the cloned capture process to begin the merge.
			If INFINITE is specified, then automatic merging starts immediately.



Table 45-18 (Cont.) Capture Process Parameters

Parameter Name	Possible Values	Default	Description
message_limit	A positive integer or INFINITE	INFINITE	The capture process stops after capturing the specified number of messages.
			If INFINITE, then the capture process continues to run regardless of the number of messages captured.
<pre>message_tracking_f requency</pre>	0 or a positive integer	2000000	The frequency at which messages captured by the capture process are tracked automatically.
			For example, if this parameter is set to the default value of 2000000, then every two-millionth message is tracked automatically.
			The tracking label used for automatic message tracking is capture_process_name: AUTOTRACK, where capture_process_name is the name of the capture process. Only the first 20 bytes of the capture process name are used; the rest is truncated if it exceeds 20 bytes.
			If 0 (zero), then no messages are tracked automatically.
parallelism	A positive integer	1	The number of preparer servers that can concurrently mine the redo log for the capture process.
			A capture process consists of one reader server, one or more preparer servers, and one builder server. The preparer servers concurrently format changes found in the redo log into logical change records (LCRs). Each reader server, preparer server, and builder server is a process, and the number of preparer servers equals the number specified for the parallelism capture process parameter. So, if parallelism is set to 5, then a capture process uses a total of seven processes: one reader server, five preparer servers, and one builder server.
			Setting the parallelism parameter to a number higher than the number of available parallel execution servers might disable the capture process. Ensure that the PROCESSES initialization parameter is set appropriately when you set the parallelism capture process parameter.
			<b>Note:</b> When you change the value of this parameter, the capture process is stopped and restarted automatically.
<pre>skip_autofiltered_ table_ddl</pre>	Y or N	Y	If Y, then the capture process does not capture data definition language (DDL) changes to tables that are automatically filtered by the capture process.
			If $\mathbb{N}$ , then the capture process can capture DDL changes to tables that are automatically filtered by the capture process if the DDL changes satisfy the capture process rule sets.



Table 45-18 (Cont.) Capture Process Parameters

Parameter Name	Possible Values	Default	Description
split_threshold	0, a positive integer, or INFINITE	1800	The amount of time, in seconds, that a stream is broken before the stream is automatically split from other streams that flow from the capture process. When a stream is split, the capture process, queue, and propagation are cloned.
			In this case, a stream is a flow of logical change records (LCRs) that flows from a capture process to an apply. A stream is broken when LCRs captured by the capture process cannot reach the apply process. For example, a stream is broken when the relevant propagation or apply process is disabled.
			This parameter is relevant only when changes captured by the capture process are applied by two or more apply processes.
			If 0 (zero), then automatic splitting starts immediately.
			If INFINITE, then automatic splitting is disabled. The stream is not split regardless of the amount of time that it is broken.
			This parameters is designed to be used with the merge_threshold parameter.
startup_seconds	0, a positive integer, or INFINITE	0	The maximum number of seconds to wait for another instantiation of the same capture process to finish. If the other instantiation of the same capture process does not finish within this time, then the capture process does not start. This parameter is useful only if you are starting the capture process manually.
			If INFINITE, then the capture process does not start until another instantiation of the same capture process finishes.
time_limit	A positive integer or INFINITE	INFINITE	The capture process stops as soon as possible after the specified number of seconds since it started.
			If INFINITE, then the capture process continues to run until it is stopped explicitly.
trace_level	0 or a positive integer	0	Set this parameter only under the guidance of Oracle Support Services.



Table 45-18 (Cont.) Capture Process Parameters

Parameter Name	Possible Values	Default	Description
use_rac_service	Y or N	Y if Replication  N if XStream	If Y, then the capture process is run in the owning instance of its queue. If N, then the client specifies where the capture process is to run.
			Note: This parameter is intended for XStream. Do not use or attempt to set this parameter in an Oracle Replication environment unless XStream optimizations are enabled by the DBMS_XSTREAM_ADM.ENABLE_GG_XSTREAM_FO R_STREAMS procedure. See "ENABLE_GG_XSTREAM_FOR_STREAMS Procedure" for information about enabling XStream optimizations.
			<b>See Also:</b> "Usage Notes" for more information about this parameter
write_alert_log	Y or N	Y	If Y, then the capture process writes a message to the alert log on exit.
			If $\mathbb{N}$ , then the capture process does not write a message to the alert log on exit.
			The message specifies the reason the capture process stopped.
<pre>xout_client_exists</pre>	process s LCRs to XStream outbound N if the ca	Y if the capture process sends	Y indicates that the capture process sends LCRs to one or more XStream outbound servers.
		XStream	N indicates that the capture process sends LCRs to one or more Oracle Apply processes.
		outbound servers  N if the capture	A single capture process cannot send LCRs to both outbound servers and apply processes.
		process sends LCRs to Oracle Apply processes	In an XStream configuration where an outbound server runs on a different database than its capture process, set this parameter to Y to enable the capture process to send LCRs to the outbound server.
			<b>Note:</b> Using XStream requires purchasing a license for the Oracle GoldenGate product. See <i>Oracle Database XStream Guide</i> .

The following usage notes apply to the SET PARAMETER procedure:

- Delays Are Possible Before New Parameter Settings Take Effect
- Parameters Interpreted as Positive Integers
- Parameters with a System Change Number (SCN) Setting
- · Parameters that Require XStream Optimizations
- XStream or Oracle GoldenGate Integrated Capture Configurations

Delays Are Possible Before New Parameter Settings Take Effect

When you alter a parameter value, a short amount of time might pass before the new value for the parameter takes effect.

Parameters Interpreted as Positive Integers

For all parameters that are interpreted as positive integers, the maximum possible value is 4,294,967,295. Where applicable, specify INFINITE for larger values.

Parameters with a System Change Number (SCN) Setting

For parameters that require an SCN setting, any valid SCN value can be specified.

Parameters that Require XStream Optimizations

A capture process uses the following parameters only when the capture process is sending logical change records (LCRs) to an XStream outbound server or when XStream optimizations are enabled for Oracle Replication components:

- capture\_idkey\_objects
- · capture sequence nextval
- excludetag
- excludetrans
- excludeuser
- excludeuserid
- getapplops
- getreplicates
- · include objects
- inline lob optimization
- max\_sga\_size
- use rac services

The <code>DBMS\_XSTREAM\_ADM.ENABLE\_GG\_XSTREAM\_FOR\_STREAMS</code> procedure enables XStream optimizations for Oracle Replication. When XStream optimizations are not enabled by the <code>DBMS\_XSTREAM\_ADM.ENABLE\_GG\_XSTREAM\_FOR\_STREAMS</code> procedure, a capture process raises an error if one of these parameters is set to any value other than its default value.

When XStream optimizations are enabled for Oracle Replication and the <code>capture\_idkey\_objects</code> parameter is set to Y, a capture process can capture ID key LCRs. ID key LCRs do not contain all of the columns for a row change. Instead, they contain the rowid of the changed row, a group of key columns to identify the row in the table, and the data for the scalar columns of the table that are supported by capture processes. An apply process can apply these changes using the information available the ID key LCRs.

To determine the database objects for which a capture process will capture ID key LCRs, run the following query on the source database:

```
SELECT OWNER, OBJECT_NAME
FROM DBA_XSTREAM_OUT_SUPPORT_MODE
WHERE SUPPORT MODE='ID KEY';
```



Using XStream requires purchasing a license for the Oracle GoldenGate product. See *Oracle Database XStream Guide*.

### See Also:

Oracle Database XStream Guide for more information about ID key LCRs

XStream or Oracle GoldenGate Integrated Capture Configurations

In an XStream or Oracle GoldenGate integrated capture configuration, the following parameters control which changes are captured by a capture process:

- capture\_idkey\_objects
- capture\_sequence\_nextval
- excludetag
- excludetrans
- excludeuser
- excludeuserid
- getapplops
- getreplicates
- include\_objects
- inline lob optimization
- max sga size
- use rac services

You can set these parameters to avoid change cycling. Change cycling sends a change back to the database where it originated. Typically, change cycling should be avoided in a replication environment so that the same change is not made to a database more than once.

In an XStream or Oracle GoldenGate integrated capture configuration that performs bidirectional replication, a GoldenGate Replicat process runs on the source database for a capture process. Therefore, the changes made by the GoldenGate Replicat are recorded in the redo log.

If an integrated configuration performs bi-directional replication, then, to avoid change cycling, the capture process should not capture the changes made by the Oracle GoldenGate Replicat process. To accomplish this goal, use the default settings for the <code>getapplops</code> and <code>getreplicates</code> parameters and exclude changes made by the user running the Replicat process. To exclude these changes, specify this user in the <code>excludeuserid</code> or <code>excludeuser</code> parameter. Typically, the user running the Oracle GoldenGate Replicat process is the XStream administrator.

In some configurations, the goal might be to capture or exclude changes made by applications or by the Replicat process. For example, an intermediate database in a replication environment might capture all of the changes made to the database, including both application changes and Replicat process changes, and send these changes to a different destination database.

Table 45-19 describes the capture process behavior when at least one of the exclude parameters is non-NULL.



Table 45-19 Behavior When at Least One exclude Parameter Is Non-NULL

getapplops Setting	getreplicates Setting	Description
Y	Y	The capture process captures all DML changes.
Y	N	The capture process captures the DML changes made by the users that are not in the excludeuserid or excludeuser parameters.
		The capture process captures the DML changes that are not in the transactions in the excludetrans parameter.
		The capture process captures only the DML changes that do not have a tag that is in the excludetags parameter.
N	Y	The capture process captures only the DML changes made by the users that are in the excludeuserid or excludeuser parameters.
		The capture process captures only the DML changes that are in the transactions in the excludetrans parameter.
		The capture process captures only the DML changes that have a tag that is in the excludetags parameter.
N	N	The capture process does not capture any DML changes.

Table 45-19 describes the capture process behavior when all of the exclude parameters are set to NULL.

Table 45-20 Behavior When All exclude Parameters Are Set to NULL

getapplops Setting	getreplicates Setting	Description
Y	Y	The capture process captures all DML changes.
Y	N	The capture process captures all DML changes.
N	Y	The capture process does not capture any DML changes.
N	N	The capture process does not capture any DML changes.

See the documentation for the Oracle GoldenGate product for more information:

http://docs.oracle.com/cd/E15881\_01/index.htm



A capture process evaluates a change using these parameters before it evaluates a change using its rule sets. Therefore, a capture process can discard a change before the change is evaluated against the capture process's rule sets. Also, regardless of the settings for these parameters, a capture process captures a change only if the change satisfies the capture process's rule sets.



Oracle Database XStream Guide

# START\_CAPTURE Procedure

This procedure starts the capture process, which mines redo logs and enqueues the mined redo information into the associated queue.

The start status is persistently recorded. Hence, if the status is ENABLED, then the capture process is started upon database instance startup.

The capture process is a background Oracle process and is prefixed by  ${\tt c.}$ 

The enqueue and dequeue state of DBMS\_AQADM.START\_QUEUE and DBMS\_AQADM.STOP\_QUEUE have no effect on the start status of a capture process.

### **Syntax**

```
DBMS_CAPTURE_ADM.START_CAPTURE(
    capture name IN VARCHAR2);
```

#### **Parameters**

#### Table 45-21 START CAPTURE Procedure Parameter

Parameter	Description	
capture_name	The name of the capture process. Do not specify an owner.	
	The capture process uses LogMiner to capture changes in the redo information. A <code>NULL</code> setting is not allowed.	

#### **Usage Notes**

The capture process status is persistently recorded. Hence, if the status is ENABLED, then the capture process is started upon database instance startup. A capture process (cnnn) is an Oracle background process.

# STOP\_CAPTURE Procedure

This procedure stops the capture process from mining redo logs.

#### **Syntax**

#### **Parameters**

Table 45-22 STOP\_CAPTURE Procedure Parameters

Parameter	Description
capture_name	The name of the capture process. A ${\tt NULL}$ setting is not allowed. Do not specify an owner.
force	If TRUE, then the procedure stops the capture process as soon as possible. If the capture process cannot stop normally, then it aborts.
	If FALSE, then the procedure stops the capture process as soon as possible. If the capture process cannot stop normally, then an ORA-26672 error is returned, and the capture process might continue to run.

### **Usage Notes**

The following usage notes apply to this procedure:

- The capture process status is persistently recorded. Hence, if the status is DISABLED or ABORTED, then the capture process is not started upon database instance startup.
- A capture process is an Oracle background process with a name in the form CPnn, where nn can include letters and numbers.
- The enqueue and dequeue state of DBMS\_AQADM.START\_QUEUE and DBMS\_AQADM.STOP\_QUEUE have no effect on the stop status of a capture process.

