124 DBMS_LOGMNR_D

The DBMS_LOGMNR_D package, one of a set of LogMiner packages, contains two subprograms: the BUILD procedure and the SET_TABLESPACE procedure.

- The BUILD procedure extracts the LogMiner data dictionary to either the redo log files or to a flat file. This information is saved in preparation for future analysis of redo log files using the LogMiner tool.
- The SET_TABLESPACE procedure re-creates all LogMiner tables in an alternate tablespace.

  The **LogMiner data dictionary** consists of the memory data structures and the database tables that are used to store and retrieve information about objects and their versions. It is referred to as the **LogMiner dictionary** throughout the LogMiner documentation.

This chapter contains the following topics:

> **See Also:**
>
> *Oracle Database Utilities* for information regarding LogMiner.

DBMS_LOGMNR_D Overview

LogMiner requires a dictionary to translate object IDs into object names when it returns redo data to you.

LogMiner gives you two options for supplying the dictionary:

- Using the online catalog
- Extracting a LogMiner dictionary to the redo log files
- Extracting a LogMiner dictionary to a flat file (for non-CDBs only)

> **Note:**
>
> In previous releases, using a flat file dictionary was one means of mining the redo logs for the changes associated with a specific PDB whose data dictionary was contained within the flat file. This feature is desupported for PDBs in Oracle Database 19c, and desupported in later releases. With Oracle Database 19c and later releases, Oracle recommends that you call DBMS_LOGMNR.START_LOGMNR, and supply the system change number (SCN) or time range that you want to mine. The SCN or time range options of START_LOGMNR are enhanced to support mining of individual PDBs.

Use the BUILD procedure to extract the LogMiner dictionary to the redo log files or a flat file. If you want to specify the online catalog as the dictionary source, you do so when you start LogMiner with the DBMS_LOGMNR.START_LOGMNR package.

Use the SET_TABLESPACE procedure if you want LogMiner tables to use a tablespace other than the default SYSAUX tablespace.

> **See Also:**
>
> DBMS_LOGMNR for information on the package subprograms used in running a LogMiner session.

DBMS_LOGMNR_D Security Model

You must have the EXECUTE_CATALOG_ROLE role to use the DBMS_LOGMNR_D package.

Summary of DBMS_LOGMNR_D Subprograms

This table lists and briefly describes the DBMS_LOGMNR_D subprograms.

In a multitenant container database (CDB), some subprograms must be called from the root. There can be other differences as well. See the individual subprogram descriptions for details.

Table 124-1 DBMS_LOGMNR_D Package Subprograms

Subprogram	Description
BUILD Procedure	Extracts the LogMiner dictionary to either a flat file or one or more redo log files
SET_TABLESPACE Procedure	Re-creates all LogMiner tables in an alternate tablespace

BUILD Procedure

This procedure extracts the LogMiner data dictionary to the redo log files.

The following considerations apply to a multitenant container database (CDB) environment.

- In a CDB environment, when you extract to the redo log files, the BUILD procedure must be called from the root database. The LogMiner data dictionary for the entire CDB is extracted to the redo log files.
- You cannot add or remove PDBs from a CDB while this procedure is running.

Note:

In previous releases, using a flat file dictionary was one means of mining the redo logs for the changes associated with a specific PDB whose data dictionary was contained within the flat file. This feature is desupported for PDBs in Oracle Database 19c, and desupported in later releases. With Oracle Database 19c and later releases, Oracle recommends that you call `DBMS_LOGMNR.START_LOGMNR`, and supply the system change number (SCN) or time range that you want to mine. The SCN or time range options of `START_LOGMNR` are enhanced to support mining of individual PDBs.

Syntax

```
DBMS_LOGMNR_D.BUILD (
     dictionary_filename  IN  VARCHAR2,
     dictionary_location  IN  VARCHAR2,
     options              IN  NUMBER);
```

Parameters

Table 124-2 BUILD Procedure Parameters

Parameter	Description
`dictionary_filename`	Specifies the name of the LogMiner dictionary file.
`dictionary_location`	Specifies the directory object for the LogMiner dictionary file.
`options`	Specifies that the LogMiner dictionary is written to the redo log files (`STORE_IN_REDO_LOGS`).

Exceptions

Table 124-3 BUILD Procedure Exceptions

Exception	Description
`ORA-01302`	Dictionary build options are missing or incorrect. This error is returned under the following conditions: • If the value of the `OPTIONS` parameter is not one of the supported values (`STORE_IN_REDO_LOGS`), or is not specified • If the `STORE_IN_REDO_LOGS` option is not specified and neither the `dictionary_filename` nor the `dictionary_location` parameter is specified • If the `STORE_IN_REDO_LOGS` option is specified and either the `dictionary_filename` or the `dictionary_location` parameter is specified

Table 124-3 (Cont.) BUILD Procedure Exceptions

Exception	Description
`ORA-01308`	Initialization parameter `UTL_FILE_DIR` is not set. **Note:** In earlier releases, you used the `UTL_FILE_DIR` initialization parameter to specify a directory location. However, as of Oracle Database 18c, the `UTL_FILE_DIR` initialization parameter is desupported. It is still supported for backward compatibility, but Oracle recommends that you instead use directory objects.
`ORA-01336`	Specified dictionary file cannot be opened. This error is returned when the dictionary file is read-only.
`ORA-01308`	Dictionary directory is not set. This error is returned under the following conditions: • The specified value for the `dictionary_location` is not a directory object. • The specified value for the `dictionary_location` is a directory object that is defined to be a file path that cannot be accessed.

Usage Notes

- To extract the LogMiner dictionary to the redo log files, specify only the `STORE_IN_REDO_LOGS` option. The size of the LogMiner dictionary may cause it to be contained in multiple redo log files.

  The combinations of parameters used result in the following behavior:

  - If you do not specify any parameters, an error is returned.
  - If you do not specify a filename and location, but do specify the `STORE_IN_REDO_LOGS` option, the LogMiner dictionary is extracted to the redo log files.
  - If you specify a filename and location, as well as the `STORE_IN_REDO_LOGS` option, an error is returned.
- Ideally, the LogMiner dictionary file is created after all database dictionary changes have been made, and before the creation of any redo log files that you want to analyze. You can use LogMiner to dump the LogMiner dictionary to the redo log files, perform DDL operations, and dynamically apply the DDL changes to the LogMiner dictionary.
- The database must be open when you run the `DBMS_LOGMNR_D.BUILD` procedure.
- To extract a LogMiner dictionary file to the redo log files, the following conditions must be met:
  - Archivelog mode must be enabled in order to generate usable redo log files.
  - The `COMPATIBLE` parameter in the initialization parameter file must be set to 9.2.0 or higher.

  - The database to which LogMiner is attached must be Oracle9i or later.

  In addition, supplemental logging (at least the minimum level) should be enabled to ensure that you can take advantage of all the features that LogMiner offers.

Examples

Example: Extracting the LogMiner Dictionary to the Redo Log Files

The following example extracts the LogMiner dictionary to the redo log files.

```
SQL> EXECUTE dbms_logmnr_d.build( -
     options => dbms_logmnr_d.store_in_redo_logs);
```

SET_TABLESPACE Procedure

This procedure moves LogMiner tables from the default SYSAUX tablespace to an alternate tablespace.

By default, all LogMiner tables are created to use the SYSAUX tablespace. However, it may be desirable to have LogMiner tables use an alternate tablespace. Use this procedure to move LogMiner tables to this alternate tablespace

In a CDB, only the LogMiner metadata in the local container is moved to the requested tablespace.

Syntax

```
DBMS_LOGMNR_D.SET_TABLESPACE (
     new_tablespace        IN VARCHAR2);
```

Parameters

Table 124-4 SET_TABLESPACE Parameter

Parameter	Description
new_tablespace	A string naming a preexisting tablespace. To move all LogMiner tables to employ this tablespace, supply this parameter.

Usage Notes

- Users upgrading from earlier versions of Oracle Database may find LogMiner tables in the SYSTEM tablespace. Oracle encourages such users to consider using the SET_TABLESPACE procedure to move the tables to the SYSAUX tablespace once they are confident that they will not be downgrading to an earlier version of Oracle Database.
- Users of this routine must supply an existing tablespace.

Example: Using the DBMS_LOGMNR_D.SET_TABLESPACE Procedure

The following example shows the creation of an alternate tablespace and execution of the DBMS_LOGMNR_D.SET_TABLESPACE procedure.

```
SQL> CREATE TABLESPACE  logmnrts$ datafile '/usr/oracle/dbs/logmnrts.f'
     SIZE 25 M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED;

SQL> EXECUTE dbms_logmnr_d.set_tablespace('logmnrts$');
```