

## DBMS\_PDB

The `DBMS_PDB` package provides an interface to examine and manipulate data about pluggable databases (PDBs) in a multitenant container database (CDB). It also contains an interface that specifies which database objects are application common objects. You can migrate a PDB from one CDB to another CDB. After the migration is complete, all backups of the PDB before the migration are available and usable in the destination CDB.



### Note:

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:

- *Oracle Database Administrator's Guide* for information about creating and managing PDBs and CDBs
- *Oracle XML DB Developer's Guide* for information about configuring protocol ports and DNS mappings
- *Oracle Database SQL Language Reference* for information about creating PDBs
- *Oracle Database Security Guide* regarding how to create audit policies in a multitenant environment

This chapter contains the following topics:

- [Overview](#)
- [Security Model](#)
- [Summary of DBMS\\_PDB Subprograms](#)

## DBMS\_PDB Overview

A multitenant container database (CDB) is an Oracle database that includes zero, one, or many user-created pluggable databases (PDBs). The `DBMS_PDB` package provides an interface to examine and manipulate data about pluggable databases (PDBs). The subprograms in this package can also set a database object to one of the following types of application common

objects in an application container: data-linked object, extended data-linked object, or metadata-linked object.

A data-linked application common object stores data in the application root that can be accessed and modified by all of the application PDBs in the application container. For an extended data-linked object, each application PDB can create its own specific data while sharing the common data in the application root. Therefore, with an extended data-linked object, only the data stored in the application root is common for all application PDBs. A metadata-linked application common object stores the metadata for the specific object, such as a table, in the application root, so that the containers in the application container have the same structure for the object but different data. This package also contains a procedure that specifies that a database object is not an application common object.

Typically, the subprograms in this package are used in the following cases:

- An application that is installed in a PDB is migrated to an application container. The application can be migrated to the application root or to an application PDB. For example, you can migrate an application installed in a PDB plugged into an Oracle Database 12c Release 1 (12.1) CDB to an application container in an Oracle Database 12c Release 2 (12.2) CDB.
- An application is installed in an application root using an installation program that does not allow setting application common objects.



#### See Also:

*Oracle Database Administrator's Guide* for information about migrating an application to an application container

## DBMS\_PDB Security Model

Users must have the `EXECUTE` privilege to run the procedures of `DBMS_PDB` package.

## Summary of DBMS\_PDB Subprograms

This table lists and describes `DBMS_PDB` package subprograms.



#### Note:

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.

**Table 146-1 DBMS\_PDB Package Subprograms**

Subprogram	Description
<a href="#">CHECK_PLUG_COMPATIBILITY Function</a>	Uses an XML file describing a pluggable database (PDB) to determine whether it may be plugged into a given multitenant container database (CDB).
<a href="#">DESCRIBE Procedure</a>	Generates an XML file describing the specified pluggable database (PDB).
<a href="#">EXPORTMANBACKUP Procedure</a>	Needs to be called only when a non-CDB is migrated as PDB. This procedure needs to be executed in non-CDB database. For PDb relocation, it is automatically called during unplugin.
<a href="#">RECOVER Procedure</a>	Generates an XML file describing a pluggable database by using data files belonging to the pluggable database (PDB).
<a href="#">REMOVE_LINK Procedure</a>	Specifies that a database object is not an application common object.
<a href="#">SET_DATA_LINKED Procedure</a>	Sets a database object to a data-linked application common object
<a href="#">SET_EXT_DATA_LINKED Procedure</a>	Sets a database object to an extended data-linked application common object.
<a href="#">SET_METADATA_LINKED Procedure</a>	Sets a database object to a metadata-linked application common object.
<a href="#">SET_PROFILE_EXPLICIT Procedure</a>	Sets a profile as an application common profile in an application container.
<a href="#">SET_ROLE_EXPLICIT Procedure</a>	Sets a role as an application common role in an application container.
<a href="#">SET_USER_EXPLICIT Procedure</a>	This procedure sets a local user as an application common user in an application container.

## CHECK\_PLUG\_COMPATIBILITY Function

This function uses an XML file describing a pluggable database (PDB) to determine whether it may be plugged into a given multitenant container database (CDB).

### Syntax

```
DBMS_PDB.CHECK_PLUG_COMPATIBILITY (  
    pdb_descr_file    IN    VARCHAR2,  
    pdb_name          IN    VARCHAR2 DEFAULT NULL)  
RETURN BOOLEAN;
```

### Parameters

**Table 146-2 CHECK\_PLUG\_COMPATIBILITY Procedure Parameters**

Parameter	Description
<code>pdb_descr_file</code>	Path of the XML file that will contain description of a PDB
<code>pdb_name</code>	Name which will be given to the PDB represented by <code>pdb_descr_file</code> when plugged into a given CDB. If not specified, the name will be extracted from <code>pdb_descr_file</code> .

## Return Values

TRUE if the PDB described by `pdb_descr_file` is compatible with the given CDB, FALSE otherwise. If this function returns FALSE, then query the `PDB_PLUG_IN_VIOLATIONS` data dictionary view to find information about the errors that are found.



### See Also:

*Oracle Database Reference* for information about the `PDB_PLUG_IN_VIOLATIONS` view

## DESCRIBE Procedure

This procedure generates an XML file describing the specified pluggable database (PDB). This file can then be passed to the `CHECK_PLUG_COMPATIBILITY` Function to determine if the PDB described by the XML file may be plugged into a given multitenant container database (CDB).

### Syntax

```
DBMS_PDB.DESCRIBE (  
    pdb_descr_file    IN    VARCHAR2,  
    pdb_name          IN    VARCHAR2 DEFAULT NULL);
```

### Parameters

**Table 146-3** DESCRIBE Procedure Parameters

Parameter	Description
<code>pdb_descr_file</code>	Path of the XML file that will contain description of a PDB
<code>pdb_name</code>	Name of a PDB to be described. A remote database is specified by including <code>@dblink</code> .

### Usage Notes

- If `pdb_name` is omitted, the PDB to which the session is connected will be described.
- If `pdb_name` is omitted, and the session is connected to the Root, an error will be returned.

### Related Topics

- [CHECK\\_PLUG\\_COMPATIBILITY Function](#)  
This function uses an XML file describing a pluggable database (PDB) to determine whether it may be plugged into a given multitenant container database (CDB).

## EXPORTRMANBACKUP Procedure

This procedure exports the RMAN backup information that belongs to a pluggable database to its dictionary before unplug so that pre-plugin backups can be used. This procedure needs to

be called only for non-CDB to PDB migration. For PDB relocation, this procedure need not be called.

**Note:**

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.

**Syntax**

```
DBMS_PDB.EXPORTMANBACKUP (  
    pdb_name      IN VARCHAR2  DEFAULT NULL);
```

**Parameters****Table 146-4 EXPORTMANBACKUP Procedure Parameters**

Parameter	Description
<code>pdb_name</code>	Name of a pluggable database whose backup information needs to be exported.  Omit this parameter if you are connected to a pluggable database or a non-cdb.

**Usage Notes**

- The PDB must be opened in `read/write` mode.
- If the database is non-CDB, then `pdb_name` must be omitted.
- If the `pdb_name` is omitted, then the pluggable database to which the session is connected is exported.
- If the `pdb_name` is omitted and the session is connected to the `root`, an error is returned.

## RECOVER Procedure

This procedure generates an XML file describing a pluggable database by using data files belonging to the pluggable database. This XML file can then be used to plug the pluggable database into a multitenant container database (CDB) using the `CREATE PLUGGABLE DATABASE` statement.

Use this procedure when an XML file describing a pluggable database is corrupted or lost.

**Syntax**

```
DBMS_PDB.RECOVER (  
    pdb_descr_file  IN  VARCHAR2,  
    pdb_name        IN  VARCHAR2,  
    filenames       IN  VARCHAR2);
```

## Parameters

**Table 146-5** *RECOVER Procedure Parameters*

Parameter	Description
<code>pdb_descr_file</code>	Path of the XML file that contains description of a pluggable database
<code>pdb_name</code>	Name of a pluggable database
<code>filenames</code>	Comma-separated list of datafile paths and/or directories containing datafiles for the pluggable database

## REMOVE\_LINK Procedure

This procedure specifies that a database object is not an application common object. In an application container, application common objects are shared between multiple containers.

### Syntax

```
DBMS_PDB.REMOVE_LINK (  
    schema_name IN VARCHAR2,  
    object_name IN VARCHAR2,  
    namespace   IN NUMBER,  
    edition_name IN VARCHAR2 DEFAULT NULL);
```

## Parameters

**Table 146-6** *REMOVE\_LINK Procedure Parameters*

Parameter	Description
<code>schema_name</code>	The name of the schema that owns the database object.
<code>object_name</code>	The name of the database object.
<code>namespace</code>	The namespace of the database object. The <code>NAMESPACE</code> column of the <code>DBA_OBJECTS</code> view shows the namespace of an object.
<code>edition_name</code>	The name of the edition for the database object.



### See Also:

*Oracle Database Administrator's Guide* for information about migrating an application to an application container

## SET\_DATA\_LINKED Procedure

This procedure sets a database object to a data-linked application common object. In an application container, data-linked application common objects store data in the application root only, and the data can be accessed by all of the application PDBs in the application container.

The data in a data-linked application common object can be modified only in the application root.

You can use this procedure to set data-linked application common objects when you migrate an application that is installed in a PDB to an application container. The application can be migrated to the application root or to an application PDB. For example, you can migrate an application installed in a PDB plugged into an Oracle Database 12c Release 1 (12.1) CDB to an application container in an Oracle Database 12c Release 2 (12.2) CDB.

### Syntax

```
DBMS_PDB.SET_DATA_LINKED (  
    schema_name  IN VARCHAR2,  
    object_name  IN VARCHAR2,  
    namespace    IN NUMBER,  
    edition_name IN VARCHAR2  DEFAULT NULL);
```

### Parameters

**Table 146-7** *SET\_DATA\_LINKED Procedure Parameters*

Parameter	Description
schema_name	The name of the schema that owns the database object.
object_name	The name of the database object.
namespace	The namespace of the database object. The <code>NAMESPACE</code> column of the <code>DBA_OBJECTS</code> view shows the namespace of an object.
edition_name	The name of the edition for the database object.



#### See Also:

*Oracle Database Administrator's Guide* for information about migrating an application to an application container

## SET\_EXT\_DATA\_LINKED Procedure

This procedure sets a database object to an extended data-linked application common object. In an application container, for an extended data-linked object, each application PDB can create its own specific data while sharing the common data in the application root. Therefore, only the data stored in the application root is common for all application PDBs.

You can use this procedure to set extended data-linked application common objects when you migrate an application that is installed in a PDB to an application container. The application can be migrated to the application root or to an application PDB. For example, you can migrate an application installed in a PDB plugged into an Oracle Database 12c Release 1 (12.1) CDB to an application container in an Oracle Database 12c Release 2 (12.2) CDB.

### Syntax

```
DBMS_PDB.SET_EXT_DATA_LINKED (  
    schema_name  IN VARCHAR2,  
    object_name  IN VARCHAR2,
```

```
namespace    IN NUMBER,  
edition_name IN VARCHAR2  DEFAULT NULL);
```

## Parameters

**Table 146-8** *SET\_EXT\_DATA\_LINKED Procedure Parameters*

Parameter	Description
schema_name	The name of the schema that owns the database object.
object_name	The name of the database object.
namespace	The namespace of the database object. The NAMESPACE column of the DBA_OBJECTS view shows the namespace of an object.
edition_name	The name of the edition for the database object.



### See Also:

*Oracle Database Administrator's Guide* for information about migrating an application to an application container

## SET\_METADATA\_LINKED Procedure

This procedure sets a database object to a metadata-linked application common object. In an application container, metadata-linked application common objects store the metadata for specific objects, such as tables, so that the containers that share the application common object have the same structure but different data.

You can use this procedure to set metadata-linked application common objects when you migrate an application that is installed in a PDB to an application container. The application can be migrated to the application root or to an application PDB. For example, you can migrate an application installed in a PDB plugged into an Oracle Database 12c Release 1 (12.1) CDB to an application container in an Oracle Database 12c Release 2 (12.2) CDB.

## Syntax

```
DBMS_PDB.SET_METADATA_LINKED (  
    schema_name  IN VARCHAR2,  
    object_name  IN VARCHAR2,  
    namespace    IN NUMBER,  
    edition_name IN VARCHAR2  DEFAULT NULL);
```

## Parameters

**Table 146-9** *SET\_METADATA\_LINKED Procedure Parameters*

Parameter	Description
schema_name	The name of the schema that owns the database object.
object_name	The name of the database object.



**Table 146-9 (Cont.) SET\_METADATA\_LINKED Procedure Parameters**

Parameter	Description
namespace	The namespace of the database object. The NAMESPACE column of the DBA_OBJECTS view shows the namespace of an object.
edition_name	The name of the edition for the database object.

**See Also:**

*Oracle Database Administrator's Guide* for information about migrating an application to an application container

## SET\_PROFILE\_EXPLICIT Procedure

This procedure sets a profile as an application common profile in an application container. This procedure is intended for migrating a profile from a previous release to an application container in the current release.

This procedure must be invoked in an application install, patch, upgrade, or uninstall operation in an application root.

### Syntax

```
DBMS_PDB.SET_PROFILE_EXPLICIT (  
    profile_name IN VARCHAR2);
```

### Parameters

**Table 146-10 SET\_PROFILE\_EXPLICIT Procedure Parameters**

Parameter	Description
profile_name	The name of the profile.

**See Also:**

*Oracle Database Administrator's Guide* for information about migrating an application to an application container.

## SET\_ROLE\_EXPLICIT Procedure

This procedure sets a role as an application common role in an application container. This procedure is intended for migrating a role from a previous release to an application container in the current release.

This procedure must be invoked in an application install, patch, upgrade, or uninstall operation in an application root.

## Syntax

```
DBMS_PDB.SET_ROLE_EXPLICIT (  
    role_name IN VARCHAR2);
```

## Parameters

**Table 146-11** *SET\_ROLE\_EXPLICIT Procedure Parameters*

Parameter	Description
role_name	The name of the role.



### See Also:

*Oracle Database Administrator's Guide* for information about migrating an application to an application container.

## SET\_USER\_EXPLICIT Procedure

This procedure sets a local user as an application common user in an application container.

This procedure must be invoked in an application install, patch, upgrade, or uninstall operation in an application root.

## Syntax

```
DBMS_PDB.SET_USER_EXPLICIT (  
    user_name IN VARCHAR2);
```

## Parameters

**Table 146-12** *SET\_USER\_EXPLICIT Procedure Parameters*

Parameter	Description
user_name	The name of the user.



### See Also:

*Oracle Database Administrator's Guide* for information about migrating an application to an application container.