

## DBMS\_EPG

The `DBMS_EPG` package implements the embedded PL/SQL gateway that enables a Web browser to invoke a PL/SQL stored procedure through an HTTP listener.

This chapter contains the following topics:

- [Overview](#)
- [Security Model](#)
- [Exceptions](#)
- [Data Structures](#)
  - [VARCHAR2\\_TABLE Table Type](#)
- [Subprogram Groups](#)
- [Summary of DBMS\\_EPG Subprograms](#)

## DBMS\_EPG Overview

The `DBMS_EPG` package is a platform on which PL/SQL users develop and deploy PL/SQL Web applications. The embedded PL/SQL gateway is an embedded version of the gateway that runs in the XML database HTTP server in the Oracle database. It provides the core features of `mod_plsql` in the database but does not require the Oracle HTTP server powered by Apache.

In order to make a PL/SQL application accessible from a browser by way of HTTP, a Database Access Descriptor (DAD) must be created and mapped to a virtual path. A DAD is a set of configuration values used for database access and the virtual path mapping makes the application accessible under a virtual path of the XML DB HTTP Server. A DAD is represented as a servlet in XML DB HTTP Server.

## DBMS\_EPG Security Model

The `XDBADMIN` role is required to invoke the configuration interface. It may be invoked by the database user "XDB".

The authorization interface can be invoked by any user.

## DBMS\_EPG Exceptions

The table in this topic lists the exceptions raised by the `DBMS_EPG` package.

**Table 84-1 DBMS\_EPG Exceptions**

Exception	Error Code	Description
DAD_NOT_FOUND	20000	Database Access Descriptor (DAD) %s not found. Ensure that the name of the DAD is correct and that it exists.

## DBMS\_EPG Data Structures

The DBMS\_EPG package defines a TABLE type.

### VARCHAR2\_TABLE Table Type

This type is used by the procedures GET\_ALL\_GLOBAL\_ATTRIBUTES, GET\_ALL\_DAD\_ATTRIBUTES, GET\_ALL\_DAD\_MAPPINGS, and GET\_DAD\_LIST to return lists of attribute names, attribute values, virtual paths, and database access descriptors (DAD).

```
TYPE VARCHAR2_TABLE IS TABLE OF VARCHAR2(4000) INDEX BY BINARY_INTEGER;
```

## DBMS\_EPG Subprogram Groups

The DBMS\_EPG consists of two interfaces: configuration subprograms and authorization subprograms.

- [Configuration Subprograms](#)
- [Authorization Subprograms](#)

## DBMS\_EPG Configuration Subprograms

The Configuration subprogram group contain the subprogram interfaces to examine and modify the global and database access descriptor (DAD) specific settings of the embedded PL/SQL gateway.

**Table 84-2 Configuration Subprogram Group**

Subprogram	Description
<a href="#">CREATE_DAD Procedure</a>	Creates a new DAD
<a href="#">DELETE_DAD_ATTRIBUTE Procedure</a>	Deletes a DAD attribute
<a href="#">DELETE_GLOBAL_ATTRIBUTE Procedure</a>	Deletes a global attribute
<a href="#">DROP_DAD Procedure</a>	Drops a DAD
<a href="#">GET_ALL_DAD_ATTRIBUTES Procedure</a>	Retrieves all the attributes of a DAD.
<a href="#">GET_ALL_DAD_MAPPINGS Procedure</a>	Retrieves all virtual paths to which the specified DAD is mapped.
<a href="#">GET_ALL_GLOBAL_ATTRIBUTES Procedure</a>	Retrieves all global attributes and values
<a href="#">GET_DAD_ATTRIBUTE Function</a>	Retrieves the value of a DAD attribute
<a href="#">GET_DAD_LIST Procedure</a>	Retrieves a list of all DADs for an Embedded Gateway instance.
<a href="#">GET_GLOBAL_ATTRIBUTE Function</a>	Retrieves the value of a global attribute
<a href="#">MAP_DAD Procedure</a>	Maps a DAD to the specified virtual path.
<a href="#">SET_DAD_ATTRIBUTE Procedure</a>	Sets the value for a DAD
<a href="#">SET_GLOBAL_ATTRIBUTE Procedure</a>	Sets the value of a global attribute
<a href="#">UNMAP_DAD Procedure</a>	Unmaps a DAD from the specified virtual path

## DBMS\_EPG Authorization Subprograms

The Authorization subprogram group contains the subprogram interfaces to authorize and deauthorize the use of a database user's privileges by the embedded PL/SQL gateway through a specific database access descriptor (DAD)

**Table 84-3 Authorization Subprogram Group**

Subprogram	Description
<a href="#">AUTHORIZE_DAD Procedure</a>	Authorizes a DAD to invoke procedures and access document tables with a database user's privileges
<a href="#">DEAUTHORIZE_DAD Procedure</a>	Deauthorizes a DAD with regard to invoking procedures and accessing document tables with a database user's privileges

## Summary of DBMS\_EPG Subprograms

This table lists the DBMS\_ALERT subprograms and briefly describes them.

**Table 84-4 DBMS\_EPG Package Subprograms**

Subprogram	Description
<a href="#">AUTHORIZE_DAD Procedure</a>	authorizes a DAD to invoke procedures and access document tables with a database user's privileges
<a href="#">CREATE_DAD Procedure</a>	Creates a new DAD
<a href="#">DEAUTHORIZE_DAD Procedure</a>	Deauthorizes a DAD with regard to invoking procedures and accessing document tables with a database user's privileges
<a href="#">DELETE_DAD_ATTRIBUTE Procedure</a>	Deletes a DAD attribute
<a href="#">DELETE_GLOBAL_ATTRIBUTE Procedure</a>	Deletes a global attribute
<a href="#">DROP_DAD Procedure</a>	Drops a DAD
<a href="#">GET_ALL_DAD_ATTRIBUTES Procedure</a>	Retrieves all the attributes of a DAD.
<a href="#">GET_ALL_DAD_MAPPINGS Procedure</a>	Retrieves all virtual paths to which the specified DAD is mapped.
<a href="#">GET_ALL_GLOBAL_ATTRIBUTES Procedure</a>	Retrieves all global attributes and values
<a href="#">GET_DAD_ATTRIBUTE Function</a>	Retrieves the value of a DAD attribute
<a href="#">GET_DAD_LIST Procedure</a>	Retrieves a list of all DADs for an Embedded Gateway instance.
<a href="#">GET_GLOBAL_ATTRIBUTE Function</a>	Retrieves the value of a global attribute
<a href="#">MAP_DAD Procedure</a>	Maps a DAD to the specified virtual path.
<a href="#">SET_DAD_ATTRIBUTE Procedure</a>	Sets the value for a DAD
<a href="#">SET_GLOBAL_ATTRIBUTE Procedure</a>	Sets the value of a global attribute
<a href="#">UNMAP_DAD Procedure</a>	Unmaps a DAD from the specified virtual path

## AUTHORIZE\_DAD Procedure

This procedure authorizes a DAD to invoke procedures and access document tables with a database user's privileges. The invoker can always authorize the use of her/his own privileges.



### See Also:

[Authorization Subprograms](#) for other subprograms in this group

### Syntax

```
DBMS_EPG.AUTHORIZE_DAD (  
    dad_name  IN  VARCHAR2,  
    path      IN  VARCHAR2 DEFAULT NULL);
```

### Parameters

**Table 84-5 AUTHORIZE\_DAD Procedure Parameters**

Parameter	Description
dad_name	The name of the DAD to create
user	The user whose privileges to deauthorize. If use, the invoker is assumed.

### Usage Notes

- To authorize the use of another user's privileges, the invoker must have the `ALTER USER` system privilege.
- The DAD must exist but its "database-username" DAD attribute does not have to be set to user to authorize.
- Multiple users can authorize the same DAD and it is up to the DAD's "database-username" setting to decide which user's privileges to use.

### Exceptions

Raises an error if the DAD or user does not exist, or the invoker does not have the needed system privilege.

### Examples

```
DBMS_EPG.AUTHORIZE_DAD('HR');
```

## CREATE\_DAD Procedure

This procedure creates a new DAD.



### See Also:

[Configuration Subprograms](#) for other subprograms in this group

### Syntax

```
DBMS_EPG.CREATE_DAD (  
    dad_name  IN  VARCHAR2,  
    path      IN  VARCHAR2 DEFAULT NULL);
```

### Parameters

**Table 84-6 CREATE\_DAD Procedure Parameters**

Parameter	Description
dad_name	The name of the DAD to create
path	The virtual path to which to map the DAD

## DEAUTHORIZE\_DAD Procedure

This procedure deauthorizes a DAD with regard to invoking procedures and accessing document tables with a database user's privileges. The invoker can always deauthorize the use of his own privileges.



### See Also:

[Authorization Subprograms](#) for other subprograms in this group

### Syntax

```
DBMS_EPG.DEAUTHORIZE_DAD (  
    dad_name  IN  VARCHAR2,  
    path      IN  VARCHAR2 DEFAULT NULL);
```

### Parameters

**Table 84-7 DEAUTHORIZE\_DAD Procedure Parameters**

Parameter	Description
dad_name	The name of the DAD for which to deauthorize use
user	The user whose privileges to deauthorize. If use, the invoker is assumed.

### Usage Notes

To deauthorize the use of another user's privileges, the invoker must have the `ALTER USER` system privilege.

### Exceptions

Raises an error if the DAD or user does not exist, or the invoker does not have the needed system privilege.

### Examples

```
DBMS_EPG.DEAUTHORIZE_DAD('HR');
```

## DELETE\_DAD\_ATTRIBUTE Procedure

This procedure deletes a DAD attribute.



#### See Also:

[Configuration Subprograms](#) for other subprograms in this group

### Syntax

```
DBMS_EPG.DELETE_DAD_ATTRIBUTE (
    dad_name      IN  VARCHAR2,
    attr_name     IN  VARCHAR2);
```

### Parameters

**Table 84-8 DELETE\_DAD\_ATTRIBUTE Procedure Parameters**

Parameter	Description
<code>dad_name</code>	The name of the DAD for which to delete a DAD attribute
<code>attr_name</code>	The name of the DAD attribute to delete

### Exceptions

Raises an error if DAD does not exist

## DELETE\_GLOBAL\_ATTRIBUTE Procedure

This procedure deletes a global attribute.



#### See Also:

[Configuration Subprograms](#) for other subprograms in this group

## Syntax

```
DBMS_EPG.DELETE_GLOBAL_ATTRIBUTE (
    attr_name    IN  VARCHAR2);
```

## Parameters

**Table 84-9 DELETE\_GLOBAL\_ATTRIBUTE Procedure Parameters**

Parameter	Description
attr_name	The global attribute to delete

## DROP\_DAD Procedure

This procedure drops a DAD. All the virtual-path mappings of the DAD will be dropped also.



### See Also:

[Configuration Subprograms](#) for other subprograms in this group

## Syntax

```
DBMS_EPG.DROP_DAD (
    dadname    IN  VARCHAR2);
```

## Parameters

**Table 84-10 DROP\_DAD Procedure Parameters**

Parameter	Description
dad_name	The DAD to drop

## Exceptions

Raises an error if the DAD does not exist.

## GET\_ALL\_DAD\_ATTRIBUTES Procedure

This procedure retrieves all the attributes of a DAD. The outputs are 2 correlated index-by-tables of the name/value pairs.



### See Also:

[Configuration Subprograms](#) for other subprograms in this group

## Syntax

```
DBMS_EPG.GET_ALL_DAD_ATTRIBUTES (
    dad_name      IN          VARCHAR2,
    attr_names    OUT NOCOPY  VARCHAR2_TABLE,
    attr_values   OUT NOCOPY  VARCHAR2_TABLE);
```

## Parameters

**Table 84-11 GET\_ALL\_DAD\_ATTRIBUTES Procedure Parameters**

Parameter	Description
dad_names	The name of the DAD
attr_names	The attribute names
attr_values	The attribute values

## Exceptions

Raises an error if DAD does not exist.

## Usage Notes

If the DAD has no attributes set, then `attr_names` and `attr_values` will be set to empty arrays.

# GET\_ALL\_DAD\_MAPPINGS Procedure

This procedure retrieves all the virtual paths to which the specified DAD is mapped.



## See Also:

[Configuration Subprograms](#) for other subprograms in this group

## Syntax

```
DBMS_EPG.GET_ALL_DAD_MAPPINGS (
    dad_name      IN          VARCHAR2,
    paths         OUT NOCOPY  VARCHAR2_TABLE);
```

## Parameters

**Table 84-12 GET\_ALL\_DAD\_MAPPINGS Procedure Parameters**

Parameter	Description
dad_names	The name of the DAD
paths	The virtual paths to which h the DAD is mapped

## Exceptions

Raises an error if DAD does not exist.



### Usage Notes

If the DAD is not mapped to any virtual path, `paths` will be set to empty arrays.

## GET\_ALL\_GLOBAL\_ATTRIBUTES Procedure

This procedure retrieves all global attributes and values. The outputs are 2 correlated index-by-tables of the name/value pairs.



### See Also:

[Configuration Subprograms](#) for other subprograms in this group

### Syntax

```
DBMS_EPG.GET_ALL_GLOBAL_ATTRIBUTES (
    attr_names      OUT  NOCOPY  VARCHAR2_TABLE,
    attr_values     OUT  NOCOPY  VARCHAR2_TABLE);
```

### Parameters

**Table 84-13 GET\_ALL\_GLOBAL\_ATTRIBUTES Procedure Parameters**

Parameter	Description
<code>attr_names</code>	The global attribute names
<code>attr_values</code>	The values of the global attributes

### Usage Notes

If the gateway instance has no global attributes set, then `attr_names` and `attr_values` will be set to empty arrays.

## GET\_DAD\_ATTRIBUTE Function

This procedure retrieves the value of a DAD attribute.



### See Also:

[Configuration Subprograms](#) for other subprograms in this group

### Syntax

```
DBMS_EPG.GET_DAD_ATTRIBUTE (
    dad_name      IN  VARCHAR2,
    attr_name     IN  VARCHAR2)
RETURN VARCHAR2;
```

## Parameters

**Table 84-14 GET\_DAD\_ATTRIBUTE Function Parameters**

Parameter	Description
dad_name	The name of the DAD for which to delete an attribute
attr_name	The name of the attribute to delete

## Return values

Returns the DAD attribute value. Returns `NULL` if attribute is unknown or has not been set.

## Exceptions

Raises an error if DAD does not exist.

# GET\_DAD\_LIST Procedure

This procedure retrieves a list of all DADs for an Embedded Gateway instance.



## See Also:

[Configuration Subprograms](#) for other subprograms in this group

## Syntax

```
DBMS_EPG.GET_DAD_LIST (
    dad_names      OUT NOCOPY  VARCHAR2_TABLE);
```

## Parameters

**Table 84-15 GET\_DAD\_LIST Procedure Parameters**

Parameter	Description
dad_names	The list of all DADs

## Usage Notes

If no DADs exist then `dad_names` will be set to an empty array.

# GET\_GLOBAL\_ATTRIBUTE Function

This function retrieves the value of a global attribute.



## See Also:

[Configuration Subprograms](#) for other subprograms in this group

## Syntax

```
DBMS_EPG.GET_GLOBAL_ATTRIBUTE (  
    attr_name IN VARCHAR2)  
RETURN VARCHAR2;
```

## Parameters

**Table 84-16 GET\_GLOBAL\_ATTRIBUTE Procedure Parameters**

Parameter	Description
attr_name	The global attribute to retrieve

## Return Values

Returns the global attribute value. Returns `NULL` if attribute has not been set or is not a valid attribute.

# MAP\_DAD Procedure

This procedure maps a DAD to the specified virtual path. If the virtual path exists already, the old virtual-path mapping will be overridden.



### See Also:

[Configuration Subprograms](#) for other subprograms in this group

## Syntax

```
DBMS_EPG.MAP_DAD (  
    dad_name IN VARCHAR2,  
    path     IN VARCHAR2);
```

## Parameters

**Table 84-17 MAP\_DAD Procedure Parameters**

Parameter	Description
dad_name	The name of the DAD to map
path	The virtual path to map

## Exceptions

Raises an error if the DAD does not exist.

## SET\_DAD\_ATTRIBUTE Procedure

This procedure sets the value for a DAD.



### See Also:

[Configuration Subprograms](#) for other subprograms in this group

### Syntax

```
DBMS_EPG.SET_DAD_ATTRIBUTE (
    dad_name     IN  VARCHAR2,
    attr_name    IN  VARCHAR2,  attr_value IN  VARCHAR2);
```

### Parameters

**Table 84-18 SET\_DAD\_ATTRIBUTE Procedure Parameters**

Parameter	Description
dad_name	The name of the DAD for which to set the attribute
attr_name	The name of the attribute to set
attr_value	The attribute value to set

**Table 84-19 Mapping Between mod\_plsql and Embedded PL/SQL Gateway DAD Attributes**

mod_plsql DAD Attribute	Embedded PL/SQL Gateway DAD Attribute	Allows Multiple Occurrences	Legal Values
PlsqlAfterProcedure	after-procedure	No	String
PlsqlAlwaysDescribe Procedure	always-describe-procedure	No	Enumeration of On, Off
PlsqlAuthentication Mode	authentication-mode	No	Enumeration of Basic, SingleSignOn, GlobalOwa, CustomOwa, PerPackageOwa
PlsqlBeforeProcedure	before-procedure	No	String
PlsqlBindBucketLengths	bind-bucket-lengths	Yes	Unsigned integer
PlsqlBindBucketWidths	bind-bucket-widths	Yes	Unsigned integer
PlsqlCGIEnvironment List	cgi-environment-list	Yes	String
PlsqlCompatibilityMode	compatibility-mode	No	Unsigned integer

**Table 84-19 (Cont.) Mapping Between mod\_plsql and Embedded PL/SQL Gateway DAD Attributes**

<b>mod_plsql DAD Attribute</b>	<b>Embedded PL/SQL Gateway DAD Attribute</b>	<b>Allows Multiple Occurrences</b>	<b>Legal Values</b>
PlsqlDatabaseUsername	database-username	No	String
PlsqlDefaultPage	default-page	No	String
PlsqlDocumentPath	document-path	No	String
PlsqlDocumentProcedure	document-procedure	No	String
PlsqlDocumentTableName	document-table-name	No	String
PlsqlErrorStyle	error-style	No	Enumeration of ApacheStyle, ModplsqlStyle, DebugStyle
PlsqlExclusionList	exclusion-list	Yes	String
PlsqlFetchBufferSize	fetch-buffer-size	No	Unsigned integer
PlsqlInfoLogging	info-logging	No	Enumeration of InfoDebug
PlsqlOWADebugEnable	owa-debug-enable	No	Enumeration of On, Off
PlsqlMaxRequestsPerSession	max-requests-per-session	No	Unsigned integer
PlsqlNLSLanguage	nls-language	No	String
PlsqlPathAlias	path-alias	No	String
PlsqlPathAliasProcedure	path-alias-procedure	No	String
PlsqlRequestValidationFunction	request-validation-function	No	String
PlsqlSessionCookieName	session-cookie-name	No	String
PlsqlSessionStateManagement	session-state-management	No	Enumeration of StatelessWithResetPackageState, StatelessWithFastRestPackageState, StatelessWithPreservePackageState
PlsqlTransferMode	transfer-mode	No	Enumeration of Char, Raw
PlsqlUploadAsLongRaw	upload-as-long-raw	No	String

### Exceptions

Raises an error if DAD does not exist or the attribute is unknown.

### Usage Notes

- If `attr_name` attribute has been set before, then the old value will be overwritten with the new `attr_value` argument.

- The embedded gateway assumes default values when the attributes are not set. The default values of the DAD attributes should be sufficient for most users of the embedded gateway. `mod_plsql` users should note the following
  - The `PlsqlDatabasePassword` attribute is not needed.
  - The `PlsqlDatabaseConnectString` attribute is not needed because the embedded gateway does not support logon to external databases.

### Examples

```
DBMS_EPG.SET_DAD_ATTRIBUTE('HR', 'default-page', 'HRApp.home');
```

## SET\_GLOBAL\_ATTRIBUTE Procedure

This procedure sets the value of a global attribute.



### See Also:

[Configuration Subprograms](#) for other subprograms in this group

### Syntax

```
DBMS_EPG.SET_GLOBAL_ATTRIBUTE (
    attr_name      IN VARCHAR2,
    attr_value     IN VARCHAR2);
```

### Parameters

**Table 84-20 SET\_GLOBAL\_ATTRIBUTE Procedure Parameters**

Parameter	Description
<code>attr_name</code>	The global attribute to set
<code>attr_value</code>	The attribute value to set

**Table 84-21 Mapping Between `mod_plsql` and Embedded PL/SQL Gateway Global Attributes**

<code>mod_plsql</code> Global Attribute	Embedded PL/SQL Gateway Global Attribute	Allows Multiple Occurrences	Legal Values
<code>PlsqlLogLevel</code>	<code>log-level</code>	No	Unsigned integer
<code>PlsqlMaxParameters</code>	<code>max-parameters</code>	No	Unsigned integer

### Usage Notes

- The attribute name is case sensitive. The value may or may not be case-sensitive depending on the attribute.
- If `attr_name` attribute has been set before, then the old value will be overwritten with the new `attr_value` argument.

Exceptions


Raises an error if the attribute is unknown.

Examples

```
dbms_epg.set_global_attribute('max-parameters', '100');
```

UNMAP\_DAD Procedure

This procedure unmaps a DAD from the specified virtual path. If path is `NULL`, the procedure removes all virtual-path mappings for the DAD but keeps the DAD.

 **See Also:**

[Configuration Subprograms](#) for other subprograms in this group

Syntax

```
DBMS_EPG.UNMAP_DAD (  
    dad_name IN VARCHAR2,  
    path      IN VARCHAR2 DEFAULT NULL);
```

Parameters

Table 84-22 UNMAP\_DAD Procedure Parameters

Parameter	Description
dad_name	The name of the DAD to unmap
path	The virtual path to unmap

Usage Notes

Raises and error if the DAD does not exist.