224

DBMS_XA

The `DBMS_XA` package contains the XA/Open interface for applications to call XA interface in PL/SQL. Using this package, application developers can switch or share transactions across SQL*Plus sessions or processes using PL/SQL.

The chapter contains the following topics:

> **See Also:**
>
> *Oracle Database Advanced Application Developer's Guide* for more information about "Developing Applications with Oracle XA"

DBMS_XA Overview

These subprograms allow a PL/SQL application to define a global transaction branch ID (`XID`) and associate or disassociate the current session with the transaction branch.

Subsequently, these transaction branches may be prepared and committed by following the two-phase commit protocol. A single-phase commit protocol is also supported if only one resource manager is involved.

Interfaces are also provided for a PL/SQL application to set the timeout values for any new global transaction branches that may start with the current session.

DBMS_XA Security Model

This package is created under `SYS`. Operations provided by this package are performed under the current calling user, not under the package owner `SYS`. Any `DBMS_XA` subprogram called from an anonymous PL/SQL block is executed using the privileges of the current user. Any `DBMS_XA` subprogram called from a stored procedure is executed using the privileges of the owner of the stored procedure.

`SELECT` or `READ` privilege on `SYS.DBA_PENDING_TRANSACTIONS` is required for users who need to execute `XA_RECOVER` subprogram.

`FORCE ANY TRANSACTION` privilege is required for users who need to manipulate XA transactions created by other users.

DBMS_XA Constants

The DBMS_XA package defines several constants that can be used for specifying parameter values.

The package uses the constants shown in Table 224-1 for use in the flag field of the XA_START Function and the XA_END Function.

Table 224-1 DBMS_XA Constants for Flag Field of XA_START & XA_END Functions

Name	Type	Value	Description
TMNOFLAGS	PLS_INTEGER	00000000	Indicates no flag value is selected.
TMSUCCESS	PLS_INTEGER	UTL_RAW.CAST_TO_BINARY_INTEGER ('04000000')	Dissociates caller from transaction branch
TMJOIN	PLS_INTEGER	UTL_RAW.CAST_TO_BINARY_INTEGER ('00200000')	Caller is joining existing transaction branch.
TMSUSPEND	PLS_INTEGER	UTL_RAW.CAST_TO_BINARY_INTEGER ('02000000')	Caller is suspending, not ending, association
TMRESUME	PLS_INTEGER	UTL_RAW.CAST_TO_BINARY_INTEGER ('08000000')	Caller is resuming association with suspended transaction branch.

The DBMS_XA package uses the constants shown in Table 224-2 for Possible Return Values

Table 224-2 DBMS_XA Constants for Possible Return Values

Name	Type	Value	Description
XA_RBBASE	PLS_INTEGER	100	Inclusive lower bound of the rollback codes
XA_RBROLLBACK	PLS_INTEGER	XA_RBBASE	Rollback was caused by an unspecified reason
XA_RBCOMMFAIL	PLS_INTEGER	XA_RBBASE+1	Rollback was caused by a communication failure
XA_RBDEADLOCK	PLS_INTEGER	XA_RBBASE+2	Deadlock was detected
XA_RBINTEGRITY	PLS_INTEGER	XA_RBBASE+3	Condition that violates the integrity of the resources was detected
XA_RBOTHER	PLS_INTEGER	XA_RBBASE+4	Resource manager rolled back the transaction for an unlisted reason
XA_RBPROTO	PLS_INTEGER	XA_RBBASE+5	Protocol error occurred in the resource manager
XA_RBTIMEOUT	PLS_INTEGER	XA_RBBASE+6	transaction branch took long
XA_RBTRANSIENT	PLS_INTEGER	XA_RBBASE+7	May retry the transaction branch

Table 224-2 (Cont.) DBMS_XA Constants for Possible Return Values

Name	Type	Value	Description
XA_RBEND	PLS_INTEGER	XA_RBTRANSIENT	Inclusive upper bound of the rollback codes
XA_NOMIGRATE	PLS_INTEGER	9	Transaction branch may have been heuristically completed
XA_HEURHAZ	PLS_INTEGER	8	Transaction branch may have been heuristically completed
XA_HEURCOM	PLS_INTEGER	7	Transaction branch has been heuristically committed
XA_HEURRB	PLS_INTEGER	6	Transaction branch has been heuristically rolled back
XA_HEURMIX	PLS_INTEGER	5	Some of the transaction branches have been heuristically committed, others rolled back
XA_RETRY	PLS_INTEGER	4	Routine returned with no effect and may be re-issued
XA_RDONLY	PLS_INTEGER	3	Transaction was read-only and has been committed
XA_OK	PLS_INTEGER	0	Normal execution
XAER_ASYNC	PLS_INTEGER	-2	Asynchronous operation already outstanding
XAER_RMERR	PLS_INTEGER	-3	Resource manager error occurred in the transaction branch
XAER_NOTA	PLS_INTEGER	-4	XID is not valid
XAER_INVAL	PLS_INTEGER	-5	Invalid arguments were given
XAER_PROTO	PLS_INTEGER	-6	Routine invoked in an improper context
XAER_RMFAIL	PLS_INTEGER	-7	Resource manager unavailable
XAER_DUPID	PLS_INTEGER	-8	XID already exists
XAER_OUTSIDE	PLS_INTEGER	-9	Resource manager doing work outside global transaction

DBMS_XA Operational Notes

In compliance with the XA specification of the X/Open CAE Standard for Distributed Transaction Processing, XA_PREPARE/COMMIT/ ROLLBACK/FORGET may not be called when the transaction is still associated with the current session. Only after XA_END has been called so that there is not any transaction associated with the current session, the application may call XA_PREPARE/COMMIT/ ROLLBACK/FORGET.

XAER_PROTO error is returned from XA_PREPARE/COMMIT/ROLLBACK/FORGET if a transaction is being associated with the current session.

Prior to calling any of the package subprograms, a connection/session must have already been established to the Oracle database server backend, or a resource manager. Resource manager identifiers are not supported. If multiple resource managers are involved, multiple

connections/sessions must be pre-established to each resource manager before calling any the package subprograms. If multiple connections/sessions are established during the course of global transaction processing, the caller must ensure that all of those connections/sessions associated with a specific global transaction branch identifier (XID) are established to the same resource manager.

DBMS_XA Data Structures

The `DBMS_XA` package uses this `OBJECT` type and associated `TABLE` type.

OBJECT Types

- DBMS_XA_XID Object Type

TABLE Types

- DBMS_XA_XID_ARRAY Table Type

DBMS_XA DBMS_XA_XID Object Type

The PL/SQL XA interface allows the PL/SQL application to define a global transaction branch id (XID) and associate/disassociate the current session with the transaction branch. XID is defined as a PL/SQL object type.

> **Note:**
>
> For more information, see "Distributed Transaction Processing: The XA Specification" in the public XA Standard.

Syntax

```
TYPE DBMS_XA_XID IS OBJECT(
  formatid      NUMBER,
  gtrid         RAW(64),
  bqual         RAW(64),
  constructor function DBMS_XA_XID(
       gtrid     IN   NUMBER)
    RETURN SELF AS RESULT,
  constructor function DBMS_XA_XID (
       gtrid     IN   RAW,
       bqual     IN   RAW)
    RETURN SELF AS RESULT,
  constructor function DBMS_XA_XID(
       formatid  IN   NUMBER,
       gtrid     IN   RAW,
       bqual     IN   RAW DEFAULT HEXTORAW('00000000000000000000000000000001'))
    RETURN SELF AS RESULT)
```

Attributes

Table 224-3 DBMS_XA_XID Object Type

Attribute	Description
`formatid`	Format identifier, a number identifying different transaction managers (TM)
`gtrid`	Global transaction identifier uniquely identifying a global transaction, of which the maximum size is 64 bytes
`bqual`	Branch qualifier, of which the maximum size is 64 bytes

DBMS_XA DBMS_XA_XID_ARRAY Table Type

This type is used to define an array of `xid` that represent a list of global transaction branches.

Syntax

```
TYPE DBMS_XA_XID_ARRAY as TABLE of DBMS_XA_XID
```

Summary of DBMS_XA Subprograms

This table lists the `DBMS_XA` subprograms and briefly describes them.

Table 224-4 DBMS_XA Package Subprograms

Subprogram	Description
DIST_TXN_SYNC Procedure	Used in recovery of synchronization when utilizing Oracle Real Application Clusters (Oracle RAC)
XA_COMMIT Function	Commits the global transaction specified by `xid`
XA_END Function	Disassociates the current session from the transaction branch specified by `xid`
XA_FORGET Function	Informs the resource manager to forget about a heuristically committed or rolled back transaction branch.
XA_GETLASTOER Function	Obtains the last Oracle error code, in case of failure of previous XA calls.
XA_PREPARE Function	Prepares the transaction branch specified in `xid` for committing the transaction subsequently if possible
XA_RECOVER Function	Obtains a list of prepared or heuristically completed transaction branches from a resource manager
XA_ROLLBACK Function	Informs the resource manager to roll back work done on behalf of a transaction branch
XA_SETTIMEOUT Function	Sets the transaction timeout in seconds for the current session
XA_START Function	Associates the current session with the transaction branch specified by `xid`

DIST_TXN_SYNC Procedure

This procedure can be used to synchronize in-doubt transactions when one of the Oracle Real Application Clusters (Oracle RAC) instances fails.

Syntax

```
DBMS_XA.DIST_TXN_SYNC;
```

XA_COMMIT Function

This function commits the global transaction specified by `xid`.

Syntax

```
DBMS_XA.XA_COMMIT (
   xid       IN  DBMS_XA_XID,
   onePhase  IN  BOOLEAN)
RETURN PLS_INTEGER;
```

Parameters

Table 224-5 *XA_COMMIT Function Parameters*

Parameter	Description
`xid`	See DBMS_XA_XID Object Type
`onePhase`	If `TRUE`, apply single phase commit

Return Values

See Table 224-2. Possible return values indicating error are: `XAER_RMERR`, `XAER_RMFAIL`, `XAER_NOTA`, `XAER_INVAL`, or `XAER_PROTO`. Other possible return values include: `XA_OK`, `XA_RB*`, `XA_HEURHAZ`, `XA_HEURCOM`, `XA_HEURRB`, and `XA_HEURMIX`.

Usage Notes

- An application must not call `COMMIT`, but instead must call `XA_COMMIT` to commit the global transaction specified by `xid`. If a user needs to commit a transaction branch that is created by other users, `FORCE ANY TRANSACTION` must be granted to the user.
- If `onePhase` is `TRUE`, the resource manager should use a one-phase commit protocol to commit the work done on behalf of `xid`. Otherwise, only if all branches of the global transaction have been prepared successfully and the preceding `XA_PREPARE` call has returned `XA_OK`, should `XA_COMMIT` be called.
- The application must make a separate `XA_COMMIT` call for each of the transaction branches of the global transaction for which `XA_PREPARE` has returned `XA_OK`.
- If the resource manager did not commit the transaction and the parameter `onePhase` is set to `TRUE`, the resource manager may return one of the `XA_RB*` code. Upon return, the resource manager has rolled back the branch's work and has released all held resources.

XA_END Function

This function disassociates the current session from the transaction branch specified by `xid`.

A transaction manager calls `XA_END` when a thread of control finishes, or needs to suspend work on, a transaction branch. This occurs when the application completes a portion of its work, either partially or in its entirety (for example, before blocking on some event in order to let other threads of control work on the branch). When `XA_END` successfully returns, the calling thread of control is no longer actively associated with the branch but the branch still exists

Syntax

```
DBMS_XA.XA_END (
   xid   IN  DBMS_XA_XID,
   flag  IN  PLS_INTEGER)
RETURN PLS_INTEGER;
```

Parameters

Table 224-6 *XA_END Function Parameters*

Parameter	Description
`xid`	See DBMS_XA_XID Object Type
`flag`	See Table 224-1.

Return Values

See Table 224-2. Possible return values in error are `XAER_RMERR`, `XAER_RMFAILED`, `XAER_NOTA`, `XAER_INVAL`, `XAER_PROTO`, or `XA_RB*`.

Usage Notes

- `TMSUCCESS` or `TMSUSPEND` may be specified in flag, and the transaction branch is disassociated with the current session in detached state if the return value is `XA_OK`. `TMFAIL` is not supported. `XA_END` may be called with either `TMSUCCESS` or `TMSUSPEND` to disassociate the transaction branch identified by xid from the current session.
- `XA_OK` is returned if `XA_END` succeeds. An application must check the return value and handle error cases. Only when `XA_OK` is returned, the application should proceed for other normal operations.
- Executing a `ROLLBACK` statement without calling `XA_END` first will rollback the changes made by the current transaction. However, the transaction context is still associated with the current session until `XA_END` is called.
- Executing a `COMMIT` statement without calling `XA_END` first will result in `ORA-02089: COMMIT` is not allowed in a subordinate session.
- Executing a `COMMIT` or a `ROLLBACK` statement after `XA_END` has no effect on the transaction identified by `xid`, since this transaction is no longer associated with the current session.

XA_FORGET Function

This function informs the resource manager to forget about a heuristically committed or rolled back transaction branch.

Syntax

```
DBMS_XA.XA_FORGET (
   xid       IN  DBMS_XA_XID)
RETURN PLS_INTEGER;
```

Parameters

Table 224-7 *XA_FORGET Function Parameters*

Parameter	Description
xid	See DBMS_XA_XID Object Type

Return Values

See Table 224-2. Possible return values are XA_OK, XAER_RMERR, XAER_RMFAIL, XAER_NOTA, XAER_INVAL, or XAER_PROTO.

XA_GETLASTOER Function

This function obtains the last Oracle error code, in case of failure of previous XA calls.

Syntax

```
DBMS_XA.XA_GETLASTOER
 RETURN PLS_INTEGER;
```

Return Values

The return value carries the last Oracle error code.

XA_PREPARE Function

This function prepares the transaction branch specified in xid for committing the transaction subsequently if possible.

Syntax

```
DBMS_XA.XA_PREPARE (
   xid   IN  DBMS_XA_XID)
RETURN PLS_INTEGER;
```

Parameters

Table 224-8 *XA_PREPARE Function Parameters*

Parameter	Description
xid	See DBMS_XA_XID Object Type

Return Values

See Table 224-2. Possible return codes include: XA_OK, XA_RDONLY, XA_RB*, XAER_RMERR, XAER_RMFAIL, XAER_NOTA, XAER_INVAL, or XAER_PROTO.

Usage Notes

- If a user needs to prepare a transaction branch that is created by other users, FORCE ANY TRANSACTION must be granted to the user.
- An application must keep track of all the branches of one global transaction, and prepare each transaction branch. Only if all branches of the global transaction have been prepared successfully and XA_PREPARE has returned XA_OK, the application may proceed to call XA_COMMIT.

XA_RECOVER Function

This function obtains a list of prepared or heuristically completed transaction branches from a resource manager.

Syntax

```
DBMS_XA.XA_RECOVER
 RETURN DBMS_XA_XID_ARRAY;
```

Return Values

See DBMS_XA_XID_ARRAY Table Type

Usage Notes

- The flags TMSTARTSCAN, TMENDSCAN, TMNOFLAGS are not supported.
- The privilege SELECT ON DBA_PENDING_TRANSACTIONS must be granted to the user who needs to call XA_RECOVER.

XA_ROLLBACK Function

This function informs the resource manager to roll back work done on behalf of a transaction branch.

Syntax

```
DBMS_XA.XA_ROLLBACK (
   xid       IN  DBMS_XA_XID)
 RETURN PLS_INTEGER;
```

Parameters

Table 224-9 *XA_ROLLBACK Function Parameters*

Parameter	Description
xid	See DBMS_XA_XID Object Type

Return Values

See Table 224-2. Possible return values are: XA_OK, XA_RB*, XA_HEURHAZ, XA_HEURCOM, XA_HEURRB, or XA_HEURMIX.

Usage Notes

If a user needs to rollback a transaction branch that created by other users, the privilege FORCE ANY TRANSACTION must be granted to the user.

XA_SETTIMEOUT Function

This function sets the transaction timeout in seconds for the current session.

Syntax

```
DBMS_XA.XA_SETTIMEOUT (
   seconds  IN  PLS_INTEGER)
RETURN PLS_INTEGER;
```

Parameters

Table 224-10 *XA_SETTIMEOUT Function Parameters*

Parameter	Description
seconds	The timeout value indicates the maximum time in seconds that a transaction branch may be disassociated from the session before the system automatically aborts the transaction. The default value is 60 seconds.

Return Values

See Table 224-2. Possible return values are XA_OK, XAER_RMERR, XAER_RMFAIL, or XAER_INVAL.

Usage Notes

Only if return value is XA_OK, is the timeout value successfully set.

XA_START Function

This function associates the current session with a transaction branch specified by the xid.

Syntax

```
DBMS_XA.XA_START (
   xid   IN  DBMS_XA_XID,    flag  IN  PLS_INTEGER)  RETURN PLS_INTEGER;
```

Parameters

Table 224-11 *XA_START Function Parameters*

Parameter	Description
xid	See DBMS_XA_XID Object Type
flag	See Table 224-1.

Return Values

See Table 224-2

Usage Notes

- If TMJOIN or TMRESUME is specified in flag, the start is for joining an existing transaction branch identified by the xid. TMJOIN flag should be used when the transaction is detached with TMSUCCESS flag. TMRESUME should be used when the transaction branch is detached with TMSUSPEND flag. XA_START may be called with either flag to join an existing transaction branch.
- If TMNOFLAGS is specified in flag, and neither TMJOIN nor TMRESUME is specified, a new transaction branch is to be started. If the transaction branch specified in xid already exists, XA_START returns an XAER_DUPID error code.
- Possible return values in error include: XAER_RMERR, XAER_RMFAIL, XAER_DUPID, XAER_OUTSIDE, XAER_NOTA, XAER_INVAL, and XAER_PROTO.
- XA_OK is returned if XA_START succeeds. An application must check the return value and handle error cases. Only when XA_OK is returned, the PL/SQL application should proceed for other normal operations. Transaction stacking is not supported. If there is an active transaction associated with the current session, may not be called to start or join another transaction. XAER_PROTO will be returned if XA_START is called with an active global transaction branch associated with the session. XAER_OUTSIDE will be returned if XA_START is called with a local transaction associated with the current session.