Oracle Data Provider for .NET 23c (23.3.3) – Dev Release Documentation Supplement

Transparent Application Failover

Oracle Transparent Application Failover (TAF) is a client-side high availability feature. It enables a client to automatically reconnect to a secondary database instance if the connected primary instance fails or shuts down.

No new failover code is required to use TAF. As the name implies, the feature is transparent, meaning ODP.NET and Oracle database will manage the instance failure detection and connection re-establishment process if TAF is enabled and configured.

ODP.NET Core, managed, and unmanaged provider types all support TAF. ODP.NET Core and managed started support with ODP.NET 23.3.3. There are differences in the TAF features each provider type supports.

TAF can be configured in the database or client side. On the client side, TAF settings can be made in the Oracle connect descriptor or through ODP.NET APIs.

TAF automatically restores some or all the elements associated with active database connections. If other elements require recovery, they should be added in the application code, such as within an ODP.NET TAF callback. Here are more details about ODP.NET TAF recovery features:

- Database connections
    - TAF automatically reestablishes the ODP.NET connection using the same connection string or an alternate connection string specified for failover.
- User database sessions
    - TAF automatically logs a user in with the same login credentials as originally used. If multiple users use the same connection, then TAF automatically logs them in as they attempt to process database commands. Unfortunately, TAF cannot automatically restore other session properties, but those properties can be restored by invoking a callback function.
- Completed commands
    - If a command completed at the time of connection failure and changed the database state, then TAF does not resend the command upon reconnection. If TAF reconnects and another command may have changed the database, then TAF issues an error message to the application. This TAF feature is available from unmanaged ODP.NET, but not managed nor core drivers.
- Open cursors for results fetching
    - TAF allows applications that began fetching rows from a cursor before failover to continue fetching rows after recovery. This is called select failover. It is re-runs a SELECT

statement using the same snapshot, discarding those rows already fetched and retrieving those rows that were not fetched initially. TAF verifies that the discarded rows are those that were returned initially, or it returns an error message. This TAF feature is available from unmanaged ODP.NET, but not managed nor core drivers.

- Active transactions
  - o Any active transactions are rolled back at the time of failure. TAF cannot preserve active transactions after failover. The application instead receives an error message until a ROLLBACK command is submitted.
- Server-side program variables
  - o Server-side program variables, such as PL/SQL package states, are lost during failures. TAF cannot recover them. They can be re-initialized by making a call from the failover callback.

TAF can be configured on the client side through the FAILOVER_MODE attribute in the TNS connect descriptor.

Table: ODP.NET TAF Failover Mode Support

TAF Failover Mode	ODP.NET Core Support	Managed ODP.NET Support	Unmanaged ODP.NET Support	Description
Session Failover	Y	Y	Y	Recreates lost connections and sessions
Select Failover	N	N	Y	Replays in-progress queries

Table: ODP.NET TAF Failover Method Support

TAF Failover Method	ODP.NET Core Support	Managed ODP.NET Support	Unmanaged ODP.NET Support	Description
Basic	Y	Y	Y	Establishes connections at failover time. This option requires almost no work on the backup database server until failover.
Preconnect	N	N	Y	Pre-establishes connections on backup instance. This provides faster failover but requires the backup instance to support all connections from the primary instance.

Note: Managed ODP.NET and ODP.NET Core do not support BACKUP and TRANSACTION TAF parameters. BACKUP specifies the failover node. TRANSACTION allows the database to complete the current transaction following a recoverable error.