12

Ensuring Application Continuity

This chapter discusses the following concepts related to the Application Continuity feature of Oracle Database:

- Overview of Ensuring Application Continuity with UCP
- Configuring the Data Source for Application Continuity
- Using Connection Labeling for Application Continuity
- Using Connection Initialization Callback for Application Continuity

12.1 Overview of Ensuring Application Continuity with UCP

Oracle Database 12*c* Release 1 (12.1) introduced the Application Continuity feature that provides a general purpose, application-independent infrastructure. Application Continuity enables recovery of work from an application perspective, after the occurrence of a planned or unplanned outage that can be related to system, communication, or hardware following a repair, a configuration change, or a patch application.

For using Application Continuity, you must first configure your data source. After that, use one of the following two features for implementing Application Continuity in your applications using Universal Connection Pool (UCP):

- Using Connection Labeling for Application Continuity
- Using Connection Initialization Callback for Application Continuity

Related Topics

- Configuring the Data Source for Application Continuity
- Using Connection Labeling for Application Continuity
- Using Connection Initialization Callback for Application Continuity

12.2 Configuring the Data Source for Application Continuity

To utilize the Application Continuity (AC) feature on a UCP data source, applications must specify an Oracle JDBC driver data source with AC support, as the connection factory class. The data source depends on the Oracle JDBC driver version that you use.

- For Oracle JDBC driver 23ai, all the driver data sources support Application Continuity. Oracle recommends that you use `oracle.jdbc.datasource.impl.OracleDataSource`, although either of the older data sources, that is, `oracle.jdbc.replay.OracleDataSourceImpl` or `oracle.jdbc.pool.OracleDataSource`, also works.
- For using Oracle JDBC driver 19c or an older driver, you must use the `oracle.jdbc.replay.OracleDataSourceImpl` to support Application Continuity.

See Also:

About Configuring Oracle JDBC for Application Continuity for Java

12.3 Using Connection Labeling for Application Continuity

Connection labeling enables an application to attach arbitrary name/value pairs to a connection. The application can request a connection with the desired label from the connection pool.

Connection labeling sets the initial state for each connection request. If the application uses connection labeling or benefits from labeling connections, then a labeling callback should be registered for Application Continuity to initialize clean connections at failover.

Every time Application Continuity gets a new connection from the underlying data source, the labeling callback executes. The callback executes during normal connection check-out and also during replay. So, the state that is created at run time is exactly re-created during replay. The initialization must be idempotent.

It is legal for the callback to execute a transaction as long as the transaction completes (either it commits or rolls back) at the end of callback invocation. Application Continuity repeats any action coded within the callback implementation, including such transaction. If an outage occurs during the execution of a UCP labeling callback, then Application Continuity may execute the callback more than once as part of the replay attempt. Again, it is important for the callback actions to be idempotent.

Related Topics

- Labeling Connections in UCP

12.4 Using Connection Initialization Callback for Application Continuity

If an application cannot use connection labeling because it cannot be changed, then the connection initialization callback is provided for such an application.

When registered, the initialization callback is executed every time a connection is borrowed from the pool and at each successful reconnection following a recoverable error..

Related Topics

- About Connection Initialization Callback