



Grouping Multiple SQL Objects in a Single Logical Unit of Work

A Logical Unit of Work (LUW) is a grouped sequence of operations on a database that must be executed completely or not at all. LUWs help to achieve database integrity. If an error is encountered during the LUW, pending database changes are cancelled and the database is rolled back to the previous correct state.

This use case describes how to group multiple SQL objects in a single LUW. This means that if a roll back or commit condition occurs for any of the SQL objects, then the entire unit will be rolled back or committed accordingly.

The *sql_luw_app.zip* archive file that is provided with this use case contains a preconfigured iWay Integration Application (iIA). Before continuing, download and extract this archive file to a location on your file system, which you can then import into iWay Integration Tools (iIT).

Application Overview and Workflow

The preconfigured iIA (*SQLLUWAPP*) that is packaged in the *sql_luw_app.zip* archive file contains a File listener as the inlet of the channel (*FILESQLCH*) and a process flow (*SQLFLOW*). The File listener can receive any XML document as an input to invoke the process flow.

The process flow consists of three SQL objects and two Commit/Rollback During Flow (*com.ibi.agents.XDCommitAgent*) service objects to handle the respective *Commit* or *Roll Back* operation.

Each SQL object calls a stored procedure, which inserts a record to the Oracle database management system (DBMS). In this use case, the stored procedure generates an exception when a unique constraint violation occurs.

When any of the three SQL objects receive the exception (or any exception generated by the BEGIN/END block of the stored procedure), the corresponding service object to *Roll Back* the LUW is called, and none of the records are written to the Oracle DBMS.




Only when all three SQL objects complete the transactions successfully, the corresponding service object to *Commit* the LUW is called, and the LUW is fully committed to the Oracle DBMS.

Configuration

The key configuration steps for this use case are completed in the preconfigured iIA (*SQLLUWAPP*) that is packaged in the *sql_luw_app.zip* archive file.

To configure multiple SQL objects as part of the same LUW:

1. For the File listener that is being used in the inlet of the channel (*FILESQLCH*), set the *Listener is Transaction Manager* parameter to *true*.

Listener is Transaction Manager   

true

2. For each of the three SQL objects in the process flow (*SQLFLOW*), set the *Need Commit* parameter to *true*.

Need Commit How should a commit be performed on completion? For a select, commit is usually not required. If Transactional (true), this operation will join the local transaction if the channel acts as a transaction manager. If it is not a transaction manager, commit is managed automatically based on the statement type. If Immediate (false) commit is performed for non-select upon execution. Use Never if your own DML/TCL will manage the commit or rollback as part of the application (process flow) logic; use of a commit object will not affect this connection.

true

3. For the Commit/Rollback During Flow (*com.ibi.agents.XDCommitAgent*) service object after the SQL objects, set the *Action* parameter to *Commit* or *Roll Back*, according to the edges returned from the SQL objects.

Action	What action should be taken
	commit
	Import Register...
	Favorites
	Commit {commit}
	Roll Back {rollback}
	[iwayhome]