

FlexDeploy for Oracle Database



The Opportunity

Traditionally, migrating database object changes is a manual, tedious, and error prone process. Making changes to database structures is not like deploying updates to code - you cannot simply replace existing structures if you want to preserve the data. For example, a table cannot be dropped and recreated to look like the new design without losing existing data.

Typical Lifecycle Management of Database Changes

- Manually comparing a new table or index structure against the current database structures to identify the required alter and/or create statements that would need to be executed.
- Executing the alter and/or create statements in the development environment.
- Repeat steps above as additional changes are applied.
- Promoting the changes to the next test environment (e.g., SIT, QA) requires either manually comparing the database structures against the changes or executing the statements in the exact order in which they were executed in development. This assumes the environments were in sync to start with. Repeat this step for each additional environment to which the changes need to be migrated.

When this lifecycle is managed through manual processes, organizations deal with quality issues stemming from human error, slower project delivery, lack of visibility into what changes are deployed to which environments, and an overall lack of control. Sound familiar? The processes may be a bit different, but the challenges are similar to managing code and configuration changes.

The Solution

While application development and operations have benefited from processes and tools related to DevOps and CI/CD, database development and administration has been left behind. FlexDeploy and the Oracle Database plugin changes all of that, enabling database developers and DBAs to move at the speed demanded by business and IT leaders.

The Oracle DB plugin provides the following solution:



- Creates a baseline of the database objects at a given point in time. This baseline is a snapshot of all database structures for the schema.
- Identifies differences between environments and generates the DDL necessary to synchronize the environments by comparing baselines from the target environment and the deployed version.

- Deploys a baseline to a target environment, such as reverting to a previous version of the database structures.
- Tracks what was deployed where and when using FlexDeploy.
- Allows users to approve and schedule deployments using FlexDeploy automation.
- Utilizes the same DevOps platform to deploy all related artifacts for a given project at the same time. For example, deploy the application code, configurations, and database objects together in a release.

Example: Managing table changes through development lifecycle

Change	Source Database	Development	QA	Production
Initial setup	Create Table v1.0	None	None	None
Deploy to dev	v1.0	Create Table v1.0	None	None
Add a new col	Alter add col v1.1	v1.0	None	None
Deploy to dev	v1.1	Alter add col v1.1	None	None
Deploy to QA	v1.1	v1.1	Create table v1.1	None
Change col length	Alter modify col v1.2	v1.1	v1.1	None
Deploy to Dev	v1.2	Alter modify col v1.2	v1.1	None
Deploy to QA	v1.2	v1.2	Alter modify col v1.2	None
Deploy to Prod	v1.2	v1.2	v1.2	Create table v1.2

The Oracle Database plugin extracts the selected database structures from a source database during the build phase and creates a baseline. The source database could be a structure only database and could be something as simple as Oracle XE running on a workstation. When a schema version (or baseline) is deployed to a target environment, the Oracle Database plugin will compare the existing structures in the target to the version of the objects stored in the generated baseline. The generated and executed DDL operations for each target environment may be unique to that environment, but the result will be each environment will now match the structure of the baseline.

Supported Oracle Object Types

Sequence	Table	Index	Constraints
View	Synonym	Function	Trigger
Type	Type Body	Package	Package Body
Procedure	Columns	Index Columns	Materialized View
Materialized View Log	Queues	Queue Tables	Queue Subscribers
Transformations	Java Source		

NOTE* *Grants are also carried forward with the object*

Additional features

- Manage full schemas, specific object types within schemas, or pick and choose individual objects within the managed schemas and object types at build time.
- Optionally recompile invalid objects.
- Ability to run the deployment in simulation mode to provide output showing what statements will be generated and executed.
- Ability to provide as input a previously generated baseline ID that will allow for an environment to be reverted to that point in time from a database structure perspective.