Deployer Guide

This document describes how to use release pipelines in Azure DevOps to deploy PL/SQL changes for a work item to target environments.

Background

Developers commit database migrations with a label that matches the work item number. A build pipeline packages the changes creating a build artifact. A release pipeline is created with one or more matching labels to deploy those changes to target environments.

See the Developer Guide for details.

Projects

Each schema targeted for migrations requires its own Liquibase project.

Project	Stack	Schema
liquibase_nsbb_main	nsbb	main (nubbmes)
liquibase_nsbb_ppc	nsbb	ppc
liquibase_sheet_main	sheet	main (psimain)
liquibase_sheet_ppc	sheet	ppc

Deployment Monitoring Console

The Deployment Monitoring Console (DMC) is a web application for viewing the status and reports for Liquibase projects. Status pages show which changesets have been deployed to which environments. Changesets can be filtered by criteria like label. Reports detailed information for Liquibase operations.

Status

The status page shows all changesets and indicates which environments they have been installed in. The status page includes filters for finding changesets

DMC filter by label example

Figure 1: DMC filter by label example

DMC report changeset summary

Figure 2: DMC report changeset summary

of interest. To view only changesets labeled for a particular work item, use the work item number in filter's label field.

Reports

The reports page list detailed reports for operations like packaging, forecasting, and deploying. Reports for the most recent operations are listed first. Click on a report to view information about the operation.

Deploy reports include details about each changeset deployed. Details and outputs can be viewed by expanding the Generated SQL & Output in the lower left corner of the changeset summary card.

Release pipelines

There are separate release pipelines for nsbb and sheet. They can be accessed in the Azure DevOps Nucor-NBT NextGen project under Pipelines / Releases. Pipelines are configured to deploy changes matching one or more labels.

Creating a release

Release pipelines use artifacts from build pipelines. A new release will default to using the latest build by default. If the release page is open, it will not check for new builds. Before creating a release on a pipeline that has been open for a while, be certain to refresh the page.

On the release pipeline, press the Create Release button. Specify one or more labels in the release creation window. Multiple labels may be used separated by commas. Press the Create button to finalize creation.

Filtering releases

Releases are named based on a sequence number and the user supplied labels. Generally, the label is the work item number. To find releases matching a certain work item, use the ADO filter menu to search for the work item number.

Specify a label when creating a release

Figure 3: Specify a label when creating a release

Filtering releases by label

Figure 4: Filtering releases by label

Deploying a release to a target

Releases are made up of stages. Each stage corresponds to a target environment like dev, qa (trn), test, or prod. Click on the stage for the deploy target and then click Deploy. Add a comment if desired and click Deploy to finalize the deployment.

The pipeline will deploy changesets matching the label to the target environment. Detailed operation can be monitored in real-time by clicking on Logs. Upon completion, a detailed report can be viewed in the DMC.

Abandoning releases

Liquibase includes techniques for handling errors and mistakes. These will fix the changelog which will be available in an artifact produced by a new build. Existing releases continue to reference the older artifact and may need to be abandoned. Anytime a developer uses cleanup, replace, or ignore workflows we should evaluate if a release for the original changes should be abandoned.

To abandon a release in Azure DevOps, click the More Options button (three dots) and select Abandon.

Workflow

The existing workflow remains in place. Deliveries still exist for non-PL/SQL changes and a delivery record must be created for PL/SQL-only changes. This does not replace guidance on delivery approval, but describes how to migrations are coordinated with the existing delivery process.

1. A delivery to a target environment is approved in a work item (in Jira or ADO).
2. The deployer checks if approved deliveries are blocked in the delivery tracker.
3. (optional) Use the DMC status page to review changesets for the work item.
4. If a release for the work item does not exist, create it.
5. Use the latest release for a work item to deploy changes to the target environment.
6. Review logs and reports for the deployment.
7. (optional) Use the DMC status page to review changesets for the work item.
8. Perform other manual delivery installation actions, as required.

If there are any issues, report them in the work item.