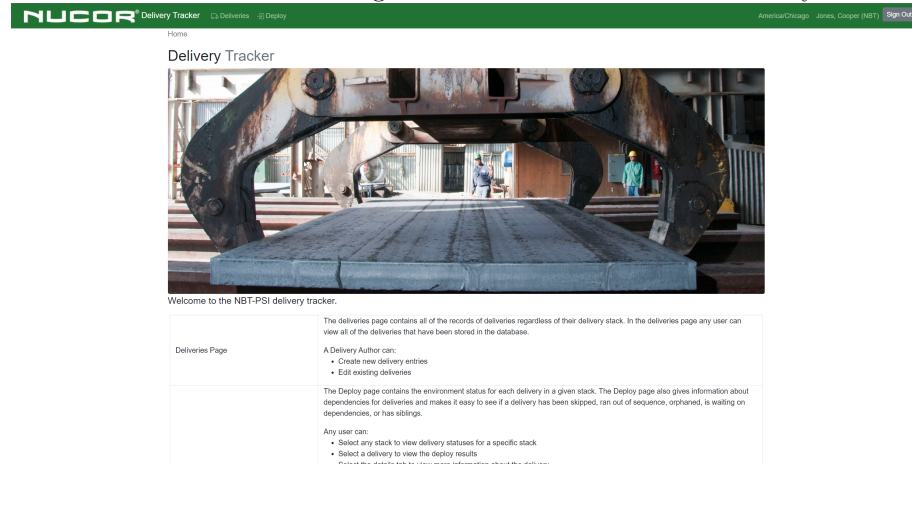# Delivery Tracker Training

This document is a training on how to utilize the Delivery Tracker.

Developer Training

Developers will have access that allows them to create and update deliveries on the Delivery Inventory page. The developers utilize the Deliveries Inventory page. The deliveries inventory page contains all of the deliveries regardless of what project they belong to.
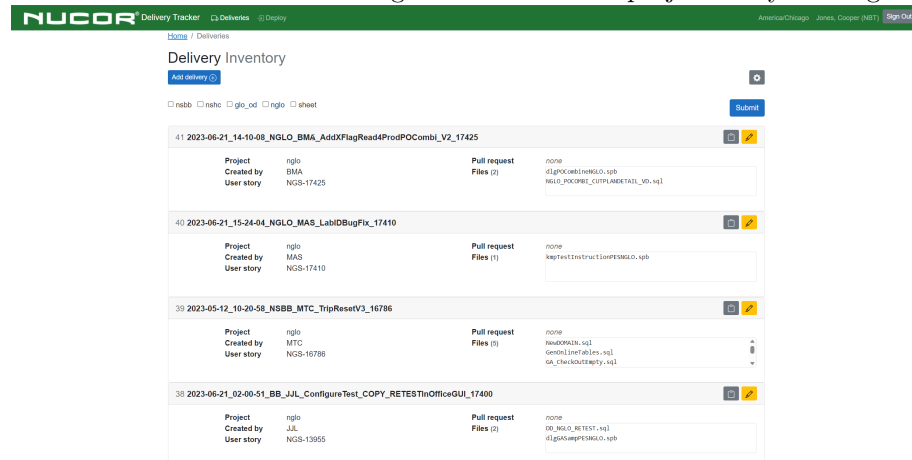
- Navigating the Delivery Inventory - When the page is loaded all deliveries are shown in descending order. - By default, each page will show four deliveries, this can be altered via filters which will be discussed in the filters section. - All deliveries that have been stored via the delivery tracker are shown by default. - Each delivery shows information about the delivery: - The project

it belongs to - Who created the delivery - The user-story that delivery was made for - The pull request that the delivery was made from - What files are present in the delivery - At the top of the delivery page there is a checkbox that shows all of the projects that have been configured. These can be used to select what projects you want to view deliveries for. These checkboxes are multi-selectable. If none of the boxes are selected all deliveries will be shown. - Filtering Deliveries - At the top right of the delivery inventory page there is a gear that is used to expand the filter options. - The filters allow a user to filter by - Name: if the delivery name contains the text in this filter, it will be shown. - User story: if the delivery is a part of this user story it will be shown. - Pull request: if the delivery is a part of this pull request it will be shown. - File name: if the delivery contains this file name it will be shown. - Created by: if the delivery was created by this developer, it will be shown. - Selecting the items per page: returns the specified number of items per page - Project: Will return the deliveries for the selected projects. If no project is selected all deliveries will be shown. - Adding a Delivery - In the top left of the Delivery Inventory page there is an Add delivery button. Once clicked it will bring up the Delivery form shown above. - The form will be used to add the delivery to the delivery tracker. It is broken into two parts. - Delivery step 1

- Name: the name of the delivery - User story: what user story this delivery was created for - Project name: selection of what project the delivery is a part of (e.g. division specific, global, etc.) - Pull request: what pull request this delivery was created from - Created by: the developer who created the delivery - Delivery actions: actions that need to be completed for the delivery to be correctly installed - Package: the delivery zip file that needs to be installed - Package detail: if the user is creating the delivery this will show none as there is no delivery that is saved. - Click save and continue this will save the zip file to the azure storage and fill out most of the rest of the form. - Delivery Step 2

The delivery has been created successfully.

Home / Deliveries / Details

Delivery step 2

Files

a.txt
b.txt

Instructions

This is a demo and has some instructions

Manual sequence dependencies

Filter

demo_delivery
test_multi
test_no_package
test demo2
demo
global_od manual dep test

Selected0

Manual simultaneous dependencies

Filter

demo_delivery
test_multi
test_no_package
test demo2
demo
global_od manual dep test

Selected0

- If there is a zip file that has been added the files will be gathered from the zip. - Instructions: - Add instructions for manual actions required to install the delivery - If there is a zip file that has been added the Instructions will be gathered from the ReadMe file - Manual sequence dependencies: these dependencies have to be installed before the delivery that is being created. - Manual simultaneous dependencies: these dependencies have to be installed at the same time as the delivery being created for the functionality to work. - In order to add a dependency simply select it from the left select box. - In order to remove a manual dependency simply select it from the right select box. - When to add manual dependencies - There is some logic to determine if a delivery has a dependency. - For sequence dependencies (deliveries that have to be installed in a sequence) if there is a prior delivery that has a file with the same name it will be a logical sequence dependency. - For simultaneous dependencies (deliveries that have to be installed at the same time for the full functionality to be present) if the deliveries have the same user story it will be a logical simultaneous dependency. - When the dependencies cannot be logically determined the user should specify manual dependencies. - Editing a Delivery - In the delivery inventory to edit a delivery the user would select the pencil in for the delivery that needs to be updated.

55 demo				📋 ⬇ ✎
Project	nsbb	**Pull request**	none	
Created by	Jones, Cooper (NBT)	**Files** (2)	a.txt	
User story	12369		b.txt	

- Follow the same process as the Adding a Delivery steps to update the delivery as necessary.

Deployment Manager Training

Deployment managers will have access that allows them to create, update and delete deploy results. This will allow for tracking deliveries through each environment in the project that the delivery exists in. The deployment manager will use the Deployment Summary page for a majority of their work.
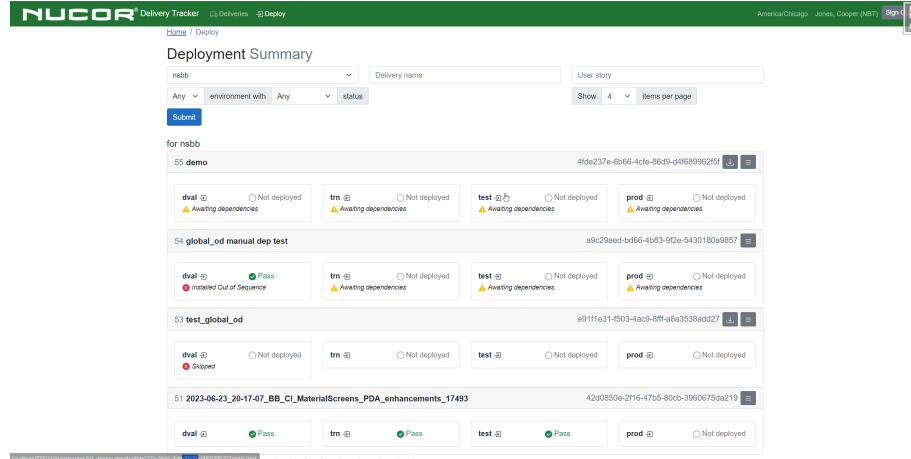
Downloading a Delivery Package With the delivery tracker enhancements one of the main features is being able to store deliveries. This will take place of the file share that we currently use for delivery drop offs and pickups. Once a developer has uploaded a deliery the package will be stored in the Azure storage where it will wait to be downloaded. - As a deployment manager the user will need to install the deliveries. In order to do that they first have to download the deliveries from the Azure storage. - This can be done on the Deployment Summary page. - When a delivery is ready to be installed by a deployment manager, they start by finding the delivery on the deployment summary page. - If the delivery has a package that is stored in Azure there will be a download button. - The deployment manger will click the download button and the zip file will be downloaded to the user's system.
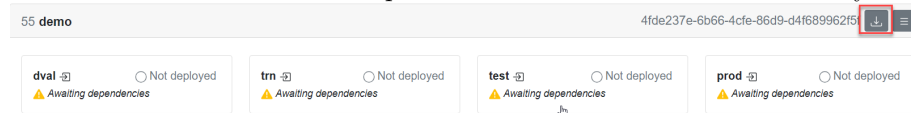
Deploy Results - After the deployment manager installs a delivery, they would navigate to the Deployment Summary page and select which delivery stack the deliveries were installed in. - The deployment manager would find the delivery that they had installed and select which environment the delivery was installed in. This brings up a Deploy Results page that contains all of the deploy results for that specific delivery for the environment selected. It will show the name of the delivery and what stack and environment the deploy result is for.

- Add a Deploy Results - To add a deploy result the user selects the Add deploy re-

sult button.

- Adding a deploy result requires the user to fill out a form. - Status: the satus is either pass or fail depending on if the delivery was successfully installed. - Deploy date: when the delivery was installed to the environment. - Deployed by: The deployment manager who installed the delivery. - Log file: Allows the deployment manager to add the delivery log from the run. - Deploy Results Information - The deploy results page gives a lot of information that aids in troubleshooting and gives general information. - General Information - Each delivery on the Deployment Summary page allows users to see the details of the delivery. - In right corner of each delivery box there is a details panel that will provide a dropdown that shows information for the delivery.

- The dropdown will show the following information: - Delivery Name - Sequence - User Story - Pull Request - Date Created - Date Added - Developer - Delivery Actions - Files - Sequence Dependencies - Simultaneous Dependencies - Instructions - Each delivery will display the status of the deployment in each of the environments: - Not deployed - Passed - Failed - Warnings and Errors - One of the main driving factors for the delivery tracker enhancements was dependency tracking - The Deployment Summary page gives visual aids to better identify dependencies and the state of the dependencies - The information the users will see is: - has siblings: means that the delivery has simultaneous dependencies - orphaned: means that one of the delivery's simultaneous dependencies was installed, but the delivery itself has not been installed yet - waiting on dependencies: means that this delivery has sequence dependencies that need to be installed before it - skipped: means that a delivery was not installed before one of its downstream sequence dependencies - ran out of sequence: means tha a delivery was installed before running all of its upstream sequence dependencies for nsbb