tracelink®

NETWORK FOR GREATER GOOD

Configuring OSE Solutions

Table of Contents

Configuring OSE Solutions

Learning Objectives

At the end of this guide, you will be able to:

- Apply lifecycle discipline (Edit, Test, Release) to manage and progress solutions.
- Use structured versioning (Patch, Minor, Major).
- Select appropriate Reuse method.
- Perform Authorization Reconciliation.
- Validate Solutions in Test.
- Verify Solution integrity and readiness before Release.

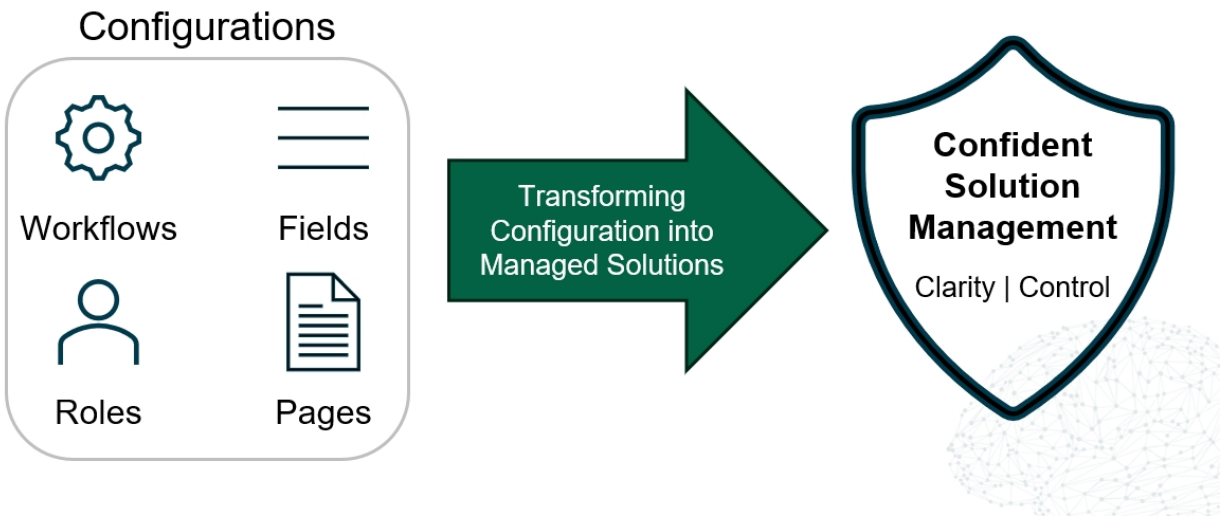
Overview

OPUS Solution Environment (OSE) provides you with a structured framework for managing and evolving solutions in a controlled, production-ready environment. You use defined lifecycle states, including Edit, Test, and Release to guide solution progression. You apply structured versioning practices, including Major, Minor, and Patch classifications, to ensure clarity and traceability. OSE also enables you to reuse solutions in a governed manner through Save As and Download capabilities, allowing consistent and controlled replication. In addition, you can rely on the platform to perform authorization reconciliation, monitor action history, and manage controlled activation through network configuration to ensure solution integrity and maintain operational stability.

Introduction

This guide shows you how to manage solutions within the OPUS Solution Environment (OSE). You may already know how to configure fields, workflows, roles, and pages. Here, you learn how to structure, validate, reuse, and progress those configurations in a structured way within OSE.

By applying these practices, you ensure clarity, consistency, and stability in solution management. The guide provides a structured framework that helps you manage solution changes effectively and responsibly within OSE.



The Problem Statement

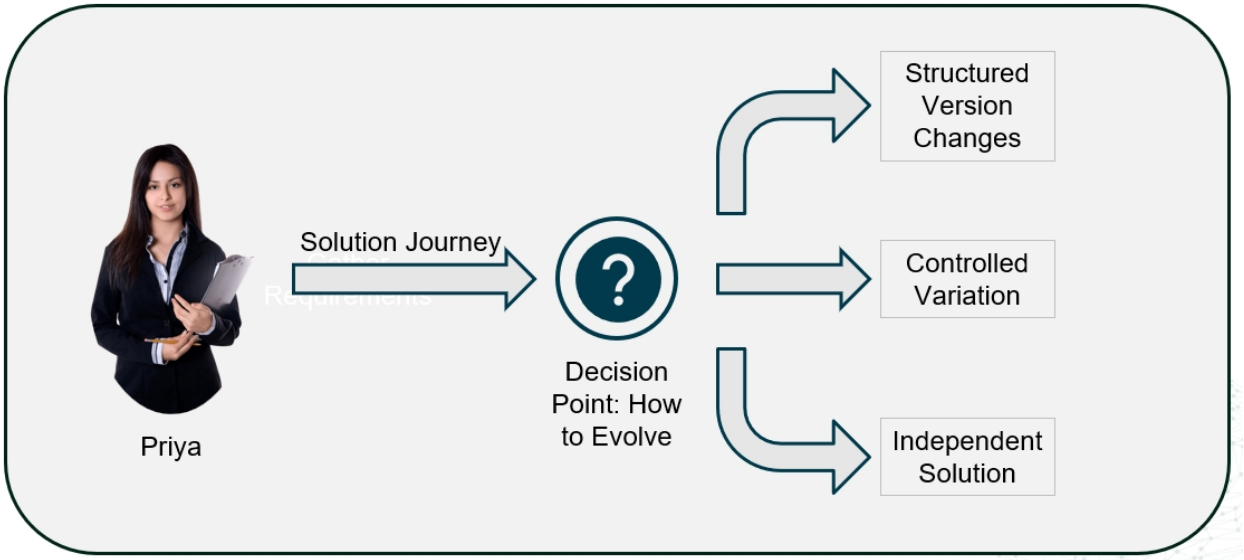
In this guide, you follow the journey of a Solution Professional working within OSE to manage Marketplace solutions built on top of the Standard solution. You take on the role of a professional who configures, versions, and maintains solutions throughout their lifecycle.

As solutions evolve, change requests arise. Not all changes are the same, and not all require the same action.

Some changes impact an existing released solution and require a formal version update to maintain release integrity and traceability. Other changes address specific business needs and require a controlled variation while maintaining alignment with the core

solution. In certain situations, requirements differ significantly and call for the creation of an independent solution to ensure clear separation and long-term scalability.

Your responsibility is to evaluate each situation and determine the appropriate path forward. The next sections explain how to assess change scenarios and apply the correct lifecycle strategy within OSE.



Why is configuration required?

Configuration is required to:

- Maintain data consistency
- Provide role-specific access and functionality
- Accelerate deployment
- Enforce business rules and compliance

You cannot rely on a one-size-fits-all setup because your partners use different terminology and your stakeholders need views aligned to their roles. You configure the system to standardize data, tailor user experiences, and accelerate implementation using no-code capabilities instead of custom development.

Lifecycle & Versioning

Lifecycle States

As your solutions evolve, you must manage updates in a controlled manner. Within OSE, you follow a defined lifecycle:

Edit -> Test -> Release

This lifecycle establishes a structured progression for each solution version:

- **Edit** - Users implement changes.
- **Test** - Users validate changes.
- **Release** - The version is approved for progression.

Lifecycle management ensures that updates remain controlled and organized as the solution grows.

NOTE: Currently, OSE only allows users to move an item from a state to next state (eg. **Edit to Test)**. Once the item is in Test, they cannot move it back to Edit. From Test, the only options are to Release the solution or create a Next Version. A future update will allow users to move items from Test back to Edit.

Versioning Strategy

The solution lifecycle defines how a version progresses through Edit → Test → Release.

You cannot modify a released version directly. When you need to make a change, you create a Next Version. This action generates a new version in the Edit state based on the current Released version.

Before creating a new version, you must assess the significance of the change. Versioning provides the structure for this decision by enabling:

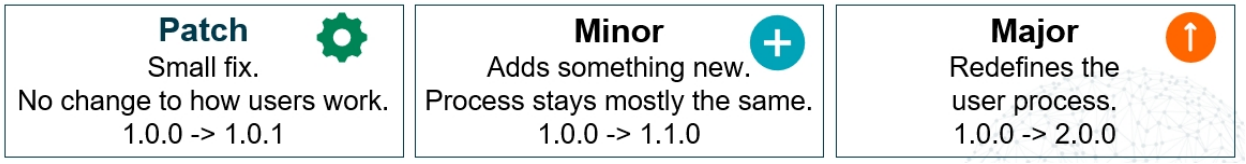
- Appropriate validation controls
- Clear version history
- Distinction between incremental updates and structural changes

OSE follows the Major.Minor.Patch convention:

- **Major** - Breaking or non-compatible changes
- **Minor** - Backward-compatible enhancements or new features
- **Patch** - Small fixes or refinements

Each new version progresses through the defined lifecycle: Edit → Test → Release.

Versioning ensures solution evolution remains structured and intentional.



To Create a New Version of a Solution

1. Log in to *opus.tracelink.com.*
2. Select **Opus Solution Environment** from the **Main Menu** .
3. In the side menu, select **Available** > **Company Solutions**.
4. On the search page, select the solution name hyperlink for the solution to be versioned.
5. Click **Next Version** from the Operations Toolbar.

6. In the Next Version push panel, select the required increment type (Major, Minor, or Patch).
7. Click **Apply**.

Strategic Reuse

Evaluating Solution Evolution Paths

As solutions evolve, scenarios such as process adjustments, additional fields, role based visibility changes, and variations of existing workflows may arise. These situations commonly occur within OSE.

The key decision is whether to update the existing solution or create a separate solution path. If the change affects all current or future users of the solution, you update the existing solution through versioning. If the change applies only to a defined customer, market, or use case and does not alter the core design, you create a controlled variation. If the change introduces requirements that differ significantly from the original structure or are expected to evolve independently, you create a separate solution to avoid cross-impact and unnecessary complexity.

For example, if a regulatory update impacts all customers, you would update the existing solution through versioning. However, if only one specific customer requires an additional workflow that does not change the core design, you would create a controlled variation. If a new market requires a fundamentally different process that will evolve separately over time, you would create a separate solution to maintain clarity and reduce complexity.

This determination represents a solution management decision, not solely a technical one.

Catalog vs Available

Before making any reuse decision, you must first identify the working layer within OSE.

The Catalog tab contains a list of all the standard, marketplace or company solutions that are available. Users can browse or search for the latest versions of solutions under this tab. It serves as a repository of all solutions that can be accessed or utilized.

The Available tab displays the solutions that have been saved or activated for a specific company. It includes solutions that have been customized or adapted for use within the

company's environment. Essentially, it reflects the solutions that are ready for deployment or use by the company.

Configuration issues commonly occur when teams assume that their work is isolated or believe that a solution is a copy when it is not. Before you use Save As, Download, or make structural changes, you must confirm the working layer.

By verifying whether the solution originates from the Catalog or Available layer, you prevent structural inconsistencies and downstream confusion.

To View Solutions in the Catalog

1. Log in to *opus.tracelink.com.*
2. Select **Opus Solution Environment** from the **Main Menu** ⁞⁞⁞.
3. In the side menu, select **Catalog** > **Standard Solutions** (or **Marketplace Solutions**, as applicable).
4. Use the Filter icon to select the relevant Application Name, if needed, and click **Apply**.
5. In the Solution Name column, select the solution hyperlink to view details.

To View Solutions in Available

1. In the side menu, select **Available** > **Company Solutions**.
2. Review the **State** column (Edit, Test, or Released).
3. Review the **Activated** column to confirm activation status.

Download Logic: Creating an Independent Copy

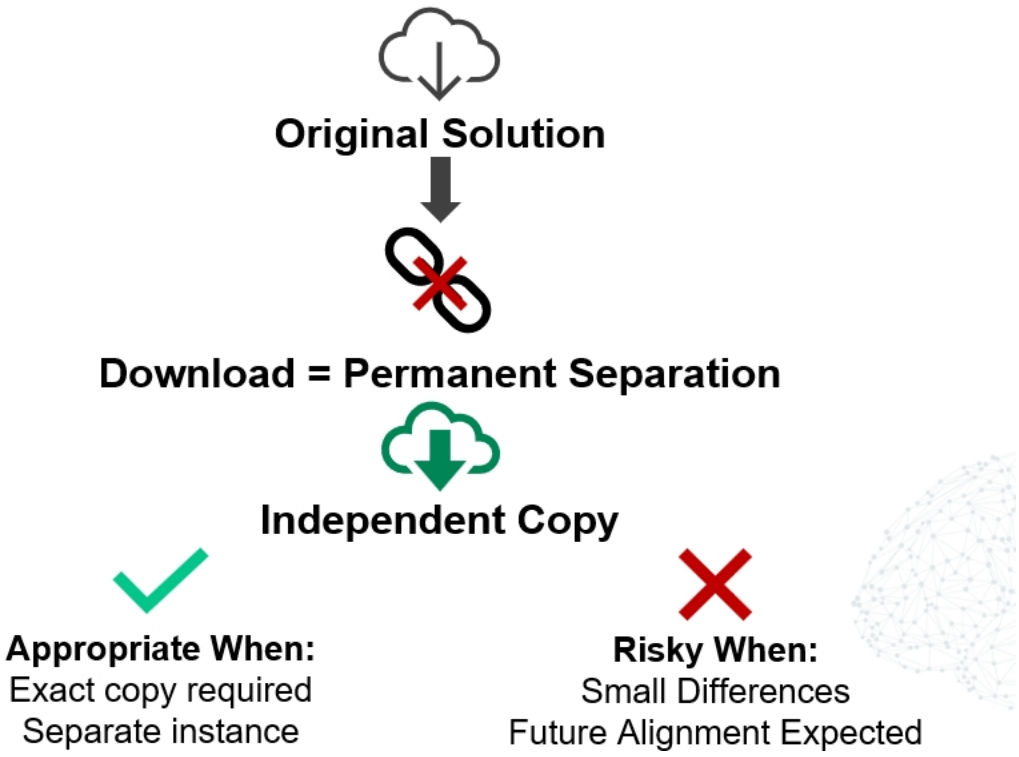
When you select Download, you create an exact copy of a Standard or Company Catalog solution in the Available section. The system places the copied solution in the Release state and keeps it inactive until you explicitly activate it.

By downloading, you create a separate solution instance based on a validated Catalog version. The new solution follows its own lifecycle, operates independently from the original, and does not receive automatic updates from the source.

In practice, you use Download when a company solution is Released in validation environment and a customer wants to Download the released solution as-is in production environment and start utilizing it.

Do not use Download for incremental adjustments. If you need to make changes after downloading, create a new version and progress it through the lifecycle.

Before you download, confirm that the source solution is in the Released state.



Save As: Copy and Configure

When you select Save As, you create a new, company-owned copy of a Standard, Marketplace, or Company solution. The system places the copied solution in the Available section, allows you to edit it immediately, and requires activation before use.

Use Save As when you need to customize the current released version of a solution from the Catalog. This action copies the released version so you can make controlled changes while keeping the original solution unchanged. For example, if you need to add an additional data field for a specific customer or internal requirement, use Save As to create your own copy and apply the modification without impacting the source solution.

This option works best when the base solution already meets most of your functional requirements and you only need targeted adjustments. Avoid using Save As if you expect the new solution to follow a completely different functional direction over time. Each copied solution requires separate version management and maintenance.

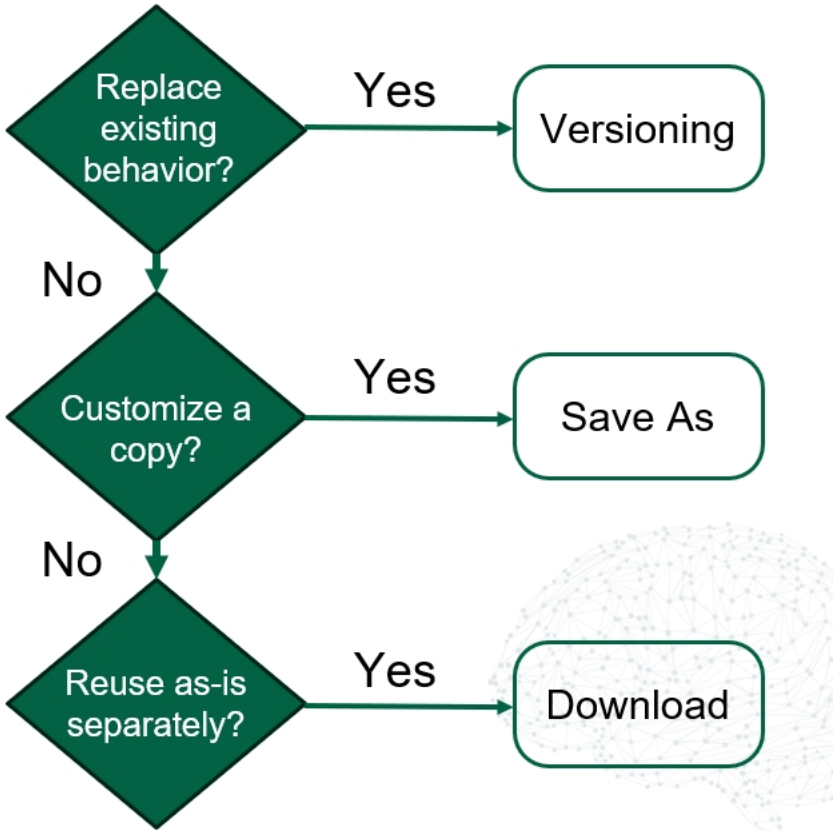
Before selecting Save As, confirm that the source solution is in the latest Release state to ensure you start from the most current baseline.

Decision

After you understand Versioning, Save As, and Download, you must determine which option aligns with the intended outcome of the change.

As a solution evolves, you must first identify the appropriate path for the update. If the change replaces prior behavior within the same solution, use Versioning to evolve it through the lifecycle. If you need customization while preserving the original released solution, use Save As to create a controlled copy for modification. If you must reuse a released solution without changes and manage it independently, use Download to create a separate instance.

Each option supports a distinct objective: evolve the same solution, copy for customization, or reuse as is under separate management. By selecting the correct path, you ensure that solution governance remains structured, organized, and predictable.



Solution Integrity

Solution Integrity: Protecting What Already Works

Solution management moves changes through the lifecycle in a controlled manner, applies the correct version classification, and reuses solutions appropriately. The next focus area is solution integrity.

Validation confirms that the solution behaves as expected within its operational context. Integrity validation ensures that the solution performs consistently across roles, workflows, and use cases. It verifies that access controls remain aligned and intentional, that action tracking functions correctly, that logic behaves consistently, and that approval flows operate as designed.

These checks strengthen confidence in the solution's readiness and occur during the Test phase before the solution progresses to Release.

Authorization Reconciliation

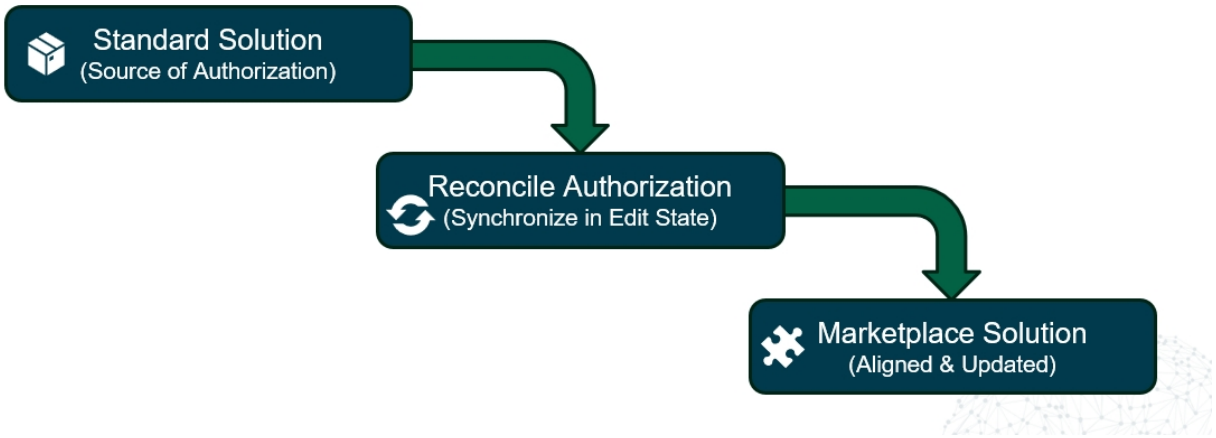
Marketplace solutions are built on top of a Standard solution and reuse its underlying configuration framework. The Standard solution defines the core structure and authorization model that Marketplace solutions reference.

When a new version of the Standard solution is released with updated authorization settings, those changes do not automatically propagate to existing Marketplace or Company solutions. In these cases, teams must perform Reconcile Authorization to maintain alignment.

Teams must reconcile when a new Standard solution version becomes active, includes authorization or policy updates, and the existing solution was created before those changes. This process ensures continued alignment with the latest Standard authorization configuration.

For example, if a new Standard version introduces updated authorization rules for a workflow event and a Marketplace solution was created before this update, reconciliation synchronizes the Marketplace solution with the active Standard authorization settings.

Reconcile Authorization updates only authorization configurations, does not modify unrelated solution components, and references the currently active Standard solution for the application. This process ensures that solutions remain aligned as the Standard solution evolves.



To Reconcile Authorizations

1. Log in to *opus.tracelink.com.*
2. Select **Opus Solution Environment** from the **Main Menu** ⁞⁞⁞.
3. In the side menu, select **Available** > **Company Solutions.**
4. In the Solution Name column, select the required solution.
5. Confirm the solution is in the **Edit** state.
6. Click **Reconcile Authorizations** from the Operations Toolbar.
7. In the Reconcile Authorizations panel, click **Apply**.

Validating Solutions in Test

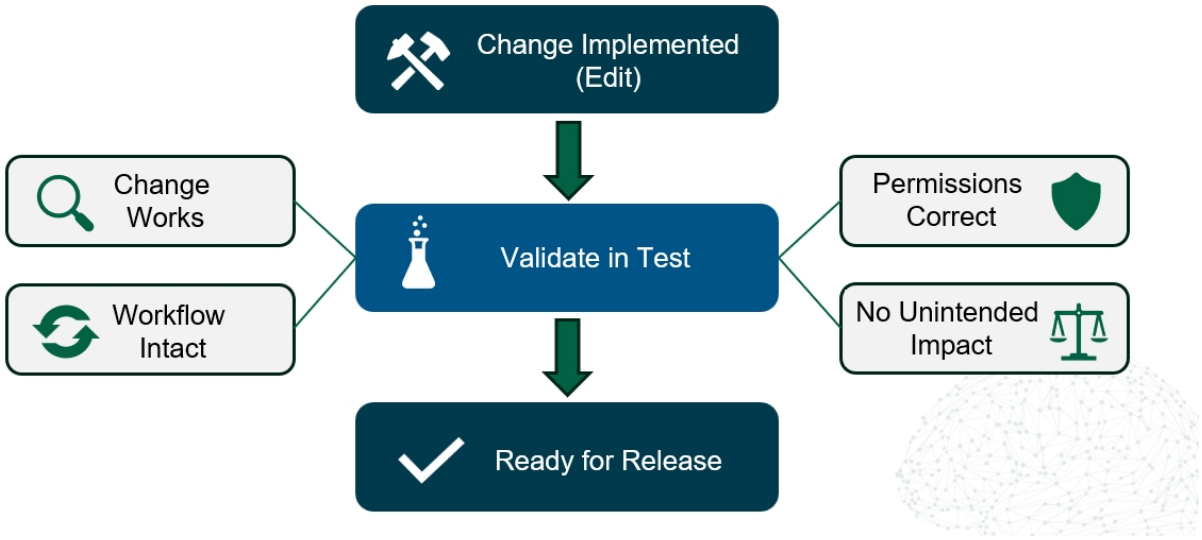
What Happens in the Test Phase?

The Test phase validates the solution and confirms readiness to progress in the lifecycle.

During this phase, the team verifies that the implemented change behaves as intended, that existing functionality continues to operate correctly, that permissions remain properly aligned, that workflow structures stay consistent, and that authorization synchronization remains intact.

Note: Solution customizations are locked (further customization prevented) when a solution is in a test phase.

The objective of the Test phase is to determine whether the version is stable and properly aligned before moving forward. A disciplined Test phase reinforces structural integrity, governance alignment, and overall solution quality, and it builds confidence before progressing to Release.



Before validating a solution in Test state, it must be activated.

To Activate a Company Solution

1. Select **Administration** from the Main Menu .

2. In the side menu, select **Network and Apps** > **Network and Apps**.
3. Click **Filter**.
4. In the Filters push panel, select your network from the Application drop-down and then click **Apply**.
5. On the Search Networks and Apps screen, select the Network hyperlink.
6. Click **Edit**.
7. In the Solution section, set the Standard Solution switch to **No**.
8. Select the company solution from the Company Solution drop-down.
9. Click **Save**.

A company solution is applied to the network.

Note: It is recommended to create a separate test process network to test the changes without impacting the solution that is already activated in the primary process network.

To Validate a Solution in Test State

1. In the side menu, select **Available** > **Company Solutions**.
2. Select the required solution and confirm the state is **Test**.
3. Select **My Network** from the Main Menu .
4. Select the network to which you have applied your Solution.
5. Click **Go**.
6. In the side menu, select **Manufacturing – Customer** > **Purchase Orders**.
7. You will be directed to the default landing page, which is the Search Page.
8. To create a new record, click **New**.

All the edits you have made, such as a renamed title, hidden assistive text, and added workflow states, will be visible on this page.

During validation, confirm the following:

- The correct version was created and the version number reflects the intended increment (Major, Minor, or Patch).
- The lifecycle state is Test, not Edit or Release.
- Workflow related changes are visible and function as expected.
- Search page updates, including page title and hidden assistive text, are correctly applied.

Action History

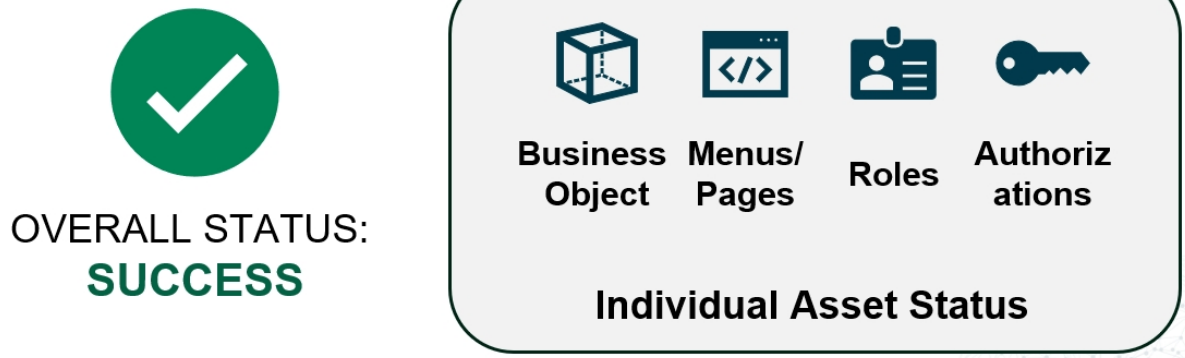
When you perform operations such as Save As, Download, or Next Version, OSE processes multiple solution components in the background. Action History shows you the progress of these processes.

It displays the operation performed, its overall status (Success, In Progress, or Failed), and the corresponding start and end times. It also breaks down the results by asset type, including Business Object Types, Menus and Page Types, Roles, Authorizations, and Bundles. Each asset type shows its individual processing status.

This visibility confirms that the system has processed all components successfully. For larger or more complex solutions, operations may take additional time; Action History allows you to monitor progress and verify completion before moving forward.

Within lifecycle management, Action History confirms that the system has successfully created a version or completed a state transition, ensuring the solution is ready for the next step.

Note: Before starting to use a solution, that has been moved to test or released, it is important to check the action history to ensure that the process has completed through as it may take some time (depending on the size of the solution), as to how long the move to test or released would take.



To Check Action History

1. Log in to *opus.tracelink.com.*
2. Select **Opus Solution Environment** from the **Main Menu** ⁞⁞⁞.
3. In the side menu, select **Available** > **Company Solutions.**
4. Select the required solution and confirm it is in **Edit** or **Test** state.
5. Navigate to the **Action History** section and expand the view.
6. Locate the relevant action in the history.
7. Verify that the overall status and each associated asset status display Success.

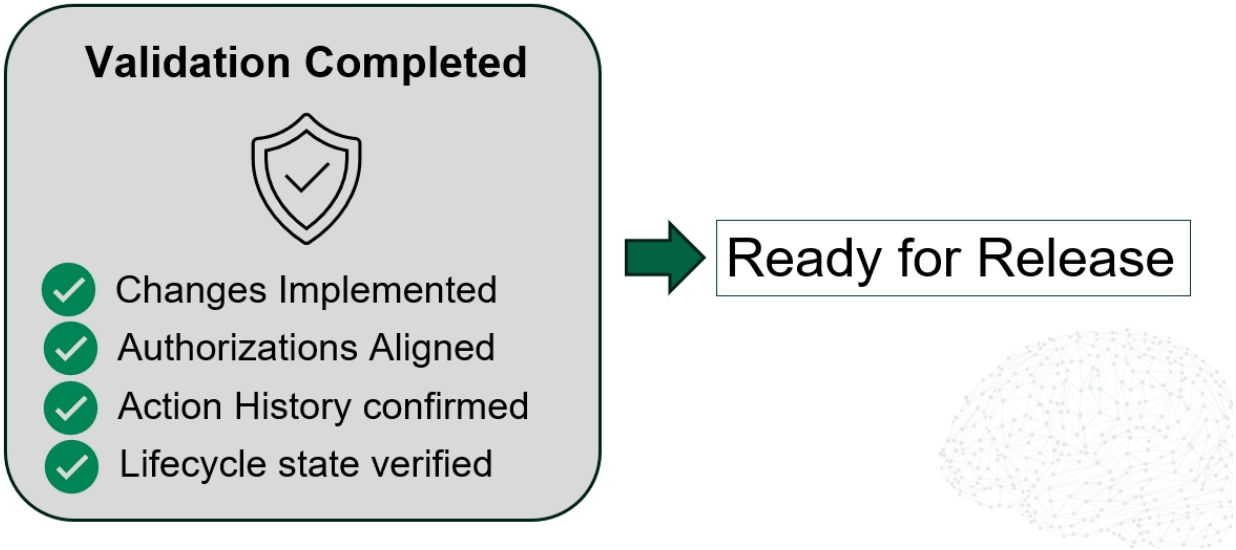
Releasing a Solution

Switch Impact Analysis

Before transitioning a solution from Test to Release, stakeholders must confirm the defined checkpoints. Release signifies that the solution is ready to progress.

At this stage, reviewers verify that all required changes have been implemented, the solution is in the correct lifecycle state, authorization alignment has been validated, Action History confirms successful processing, and the configuration reflects the intended outcome.

Structured validation must precede Release to ensure the solution remains organized, aligned, and prepared for continued evolution. Adhering to these checkpoints supports stability and clarity throughout the lifecycle.



Release the Company Solution

1. In the side menu, select **Available** > **Company Solutions**.
2. From the Solution Name column, select the link for the desired solution that is in the **Test** state.

3. Click **Edit** from the Operations toolbar.
4. Click the **Move To** button to transition the Solution from the **Test** state to the **Release** state.
5. Click **Save** in the Operations toolbar.
6. In the side menu, select **Catalog** > **Company Solutions** to display the **Search Solutions** page.

Moving Solution from Validation to Production

Solution progression from Validation to Production follows a controlled process.

In the Validation environment, users create, configure, test, and advance the solution to Release. After Release, the system publishes the solution to the Company Catalog, which makes the validated version available across environments.

In the Production environment, authorized users download the solution from the Company Catalog and activate it within the appropriate Process Network. This process ensures that only validated versions are deployed to Production.

When subsequent changes are required, users repeat the same structured process. They create a new version in Validation, progress it through Test and Release, publish it to the Company Catalog, and then download it into Production.

This approach keeps solution movement controlled, consistent, and repeatable across environments.