



## TRACELINK UNIVERSITY

**Home**  
**Resources**  
**TraceLink University**

# Workflows Streamline Business Processes With Configurability and Automation

### Purchase Order Details

View and edit purchase orders.

The diagram shows a horizontal workflow with two main stages: 'Draft' and 'Submit'. The 'Draft' stage is represented by a green circle with a person icon, and the 'Submit' stage is represented by a green circle with a checkmark icon. A teal bar labeled 'Workflow' spans across both stages.

### New Purchase Order

Add new purchase order.

The diagram shows a horizontal workflow with seven stages: Draft, Submit, Processing, Processed, Preparing To Send, Sending, and Sent. The 'Draft' stage is active, indicated by a green circle with a person icon. A 'Move To' button is positioned below the 'Submit' stage.

**Order Information**

PO Number*	260320210011	Order Date	06 Sep 2024
Delivery Term Code	INCO 1	Delivery Term Description	COD
Payment Terms By Days	14	Payment Terms By Percentage	5
Currency	USD		

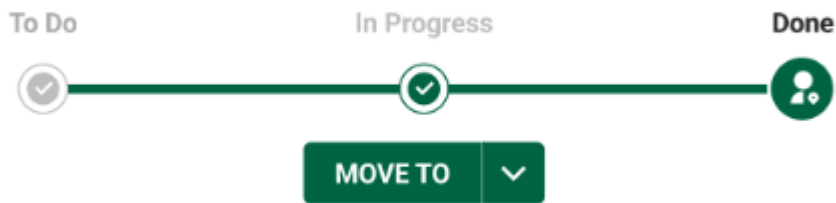
Customer and Supplier

Ship To Location and Bill To Location

Workflows are powerful mechanisms that enhance business process management by providing structure, automation, and integration. By leveraging workflows, businesses can achieve greater efficiency, consistency, and flexibility in their operations. Whether using an out-of-the-box workflow or workflows tailored to

specific business needs, the ability to define, manage, and automate processes is crucial for modern business success.

## Edit page



A **workflow** is a set of states and transitions for an object type (e.g. business object, business transaction object). Workflows facilitate business processes by defining sequences of **states** (i.e. steps or stages) and **transitions**, which may be configured to leverage **transition conditions** and **post-transition actions**.

These structured workflow elements allow for efficient, flexible, and controlled management of business processes, while leveraging the object-driven approach of Opus and thereby enhancing the reliability and flexibility of various business processes.

## Types of Workflows

There are 2 types of workflows: **standard workflows** and **business object workflows**. As you progress through the key constructs of workflows, we will offer examples that include both a Supply Chain Work Management (SCWM) business object (e.g. task) and a Multienterprise Information Network Tower (MINT) transaction object (e.g. purchase order) to illustrate the nuances with the workflow design.

### Standard Workflow

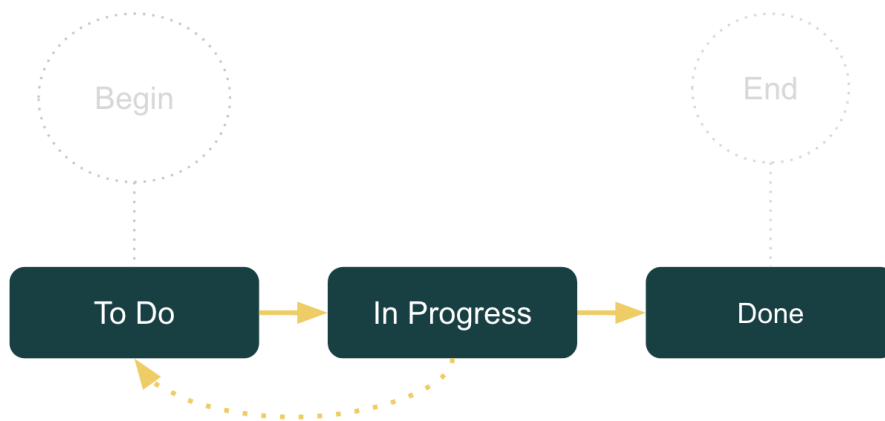
#### Standard Workflows

Standard workflows are predefined sequences of states and transitions for an object type, commonly used for business processes. Opus Developers define the standard workflow in conjunction with its standard object type.

Solution Designers *cannot* configure standard workflows; however, understanding the standard workflow helps them think about how to extend the standard workflow with configurations in their company solutions, which is covered in the next section.

## Standard Workflow Examples

### Business Object Example With Standard Workflow



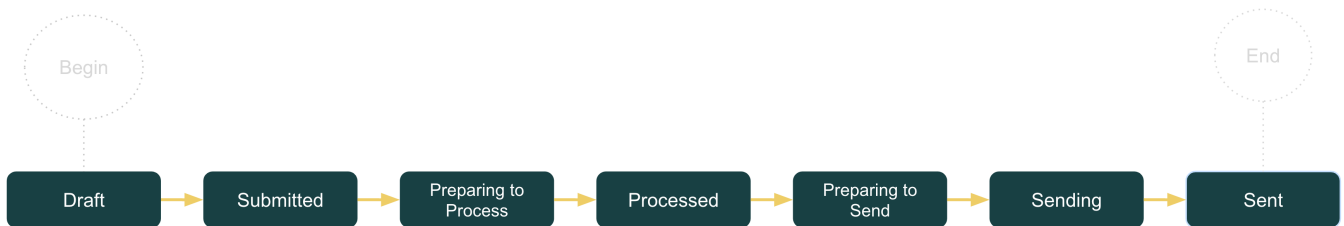
Let's take a look at SCWM's task object type. This is the standard workflow of a task:

As you can see, this is a very simple workflow where To Do, In Progress, and Done are the base states that can move between each other. Standard workflows typically transition through states via human interaction.

### Business Transaction Object (BTO) Example With Transaction Workflow

There is more robustness in workflow states for BTOs (e.g. MINT's purchase orders), as transaction workflows are designed to manage business processes where the steps are largely or entirely automated, minimizing (and even

eliminating) the need for human interaction. Transaction workflows are commonly used in orchestrated business processes, such as commerce, where purchase orders, purchase order acknowledgments, invoices, ASNs, and payments can all leverage the same workflow out of the box while configuring additional logic that can leverage persisted data across multiple BTO workflows to increase visibility within an end-to-end process.



These workflows are designed to handle routine tasks and transactions, providing a consistent and efficient design for managing business transaction objects.

### **Benefits of Standard Workflows**

- **Alignment With Business Processes** – Ensure the workflow logic closely aligns with the business processes out of the box.
- **Enhanced Productivity** – By automating complex processes, especially transactions, workflows can significantly improve productivity and reduce the potential for errors.
- **Consistency** - Standard workflows ensure consistency across all of your processes, ensuring that all of your business objects will move through the same lifecycle and limiting the possibility of human error during transitions.
- **Simplicity** - Opus Developers can leverage Opus's no-code paradigm to accelerate definition and development without the need for complex logic.

### **Exploring the Depths of Transaction Workflows**

A transaction workflow is valuable because it automates and standardizes the flow of countless business transactions, ensuring that processes follow a consistent and

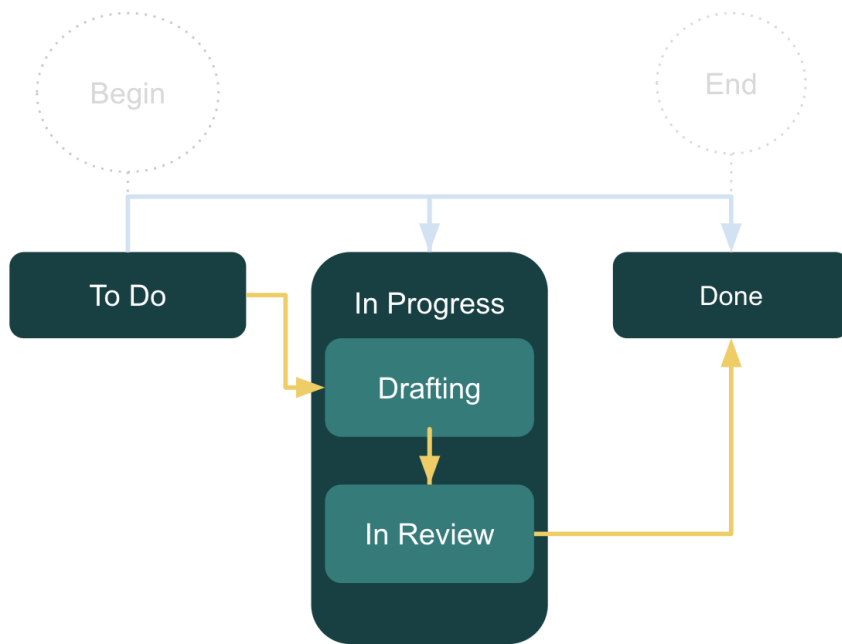
efficient path from initiation to completion. It reduces (and can even eliminate) manual effort, minimizes errors, and improves operational efficiency by enforcing predefined rules and transitions between steps.

It is also a key element in interoperability. By defining a standardized sequence of steps that business transactions follow, transaction workflows ensure that data and processes are consistent. This consistency allows diverse systems to communicate and exchange data more effectively, reducing integration complexity and ensuring smooth data exchanges across companies and systems in multienterprise environments.

Additionally, transaction workflows provide real-time visibility into the status of transactions, enabling better decision-making and quicker responses to issues, while seamlessly integrating with multiple systems and stakeholders across the supply chain.

- **Transaction Workflow** – Designed to guide a Transaction Object or BTO through a series of predefined steps using the Transaction Workflow Runtime (TWR). It can pause for user input when additional data is needed, allowing edits before continuing. The workflow is flexible, with the ability to integrate configured logic at each step of the workflow to handle both automated and manual processes.
- **Transaction Workflow Runtime (TWR)** – Automatically manages a BTO through its workflow, ensuring each step is completed as defined. It pauses when more information is needed, allowing user edits, and resumes once resolved. TWR supports configured logic for complex scenarios, combining automation with manual interventions for efficient business process execution.

## **Business Object Workflow**



Unlike standard workflows, business object workflows offer flexibility to Solution Designers by providing configuration options to meet unique requirements. Business object workflows have the ability to extend that original standard workflow through configurations on what are called substates, along with transitions.

This approach provides a flexible and effective way to handle specialized and complex business processes that standard workflows cannot fully address. By allowing businesses to define specific substates, business object workflows ensure that the workflow logic facilitates the unique needs of the organization.

### **Benefits of Business Object Workflows**

1. **Flexibility** – Allow businesses to define specific substates and transitions that are unique to their processes, providing a higher level of configuration compared to standard workflows.
2. **Specificity** – Can include detailed conditions and actions that are specific to your business context, enabling precise control over workflow execution.
3. **Integration** – Can integrate with processes, allowing for seamless data flow

and operational coherence across different business functions (see the **Workflow Subscription** section for more information).

We will illustrate how a standard workflow becomes a business object workflow by using key workflow constructs, like states, substates, and transitions, and using both business objects and BTO examples.

## **Workflow Constructs**

Now let's break down how standard workflows were assembled with the core building blocks of a workflow: states and transitions. Regardless of the type of workflow, states and transitions are needed.

### **States**

Workflow states represent an object's current position within the overall workflow, while transitions direct the object through various stages and define the permissible paths between states. Workflow states will always include base states and can be used with substates (which are configurable), representing the full spectrum of steps that an object type may occupy during its lifecycle.

### **Base States**

Base states are the primary states in a workflow that denote the phases in the lifecycle of an object type. Just like the overall standard workflow, Opus Developers create this construct. Solution Designers do not directly create or configure a standard workflow, but Solution Designers need to understand the standard workflow, as it is the foundation upon which they will extend functionality.

### **Business Object Example**

Using SCWM's task object type, you can see the simplicity and clarity of the base states used.

- **To Do**
- **In Progress**
- **Done**

#### **BTO Example**

Using MINT's Purchase Order object type, you can see a more complex example of how base states are used. All B2B transactions come with this workflow out of the box.

- **Draft**
- **Submitted**
- **Preparing To Process**
- **Processed**
- **Preparing To Send**
- **Sending**
- **Sent**

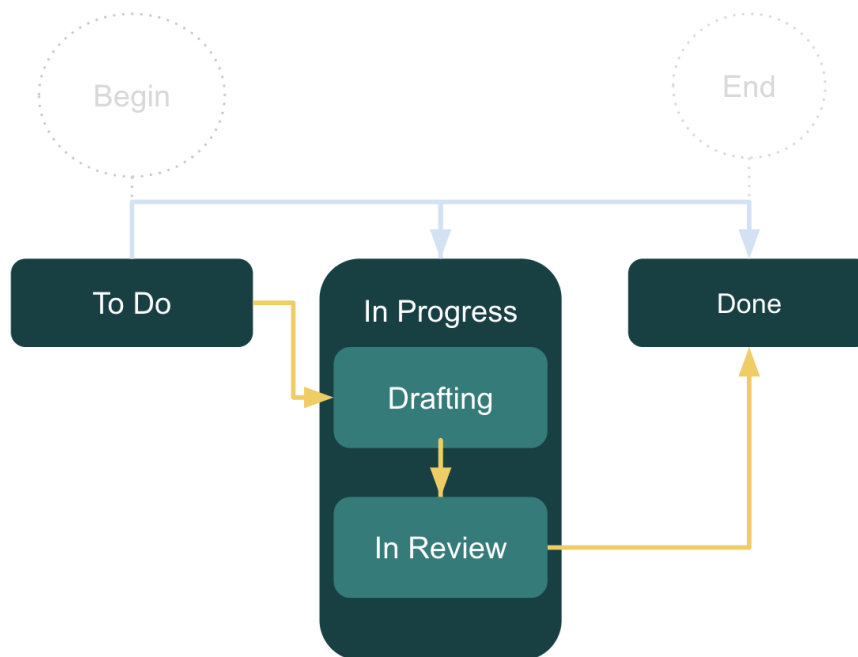
As mentioned in the object type fundamentals, BTOs are more robust, enhanced object types, with an extra focus on handling business transactions. Message flows for business transactions have historically had a shortcoming where they could not retain data for querying or viewing. Transaction object types, with the help of their workflow, offer that visibility that was previously lacking.

#### **Substates**

Substates are more granular states that exist within a base state. They provide additional detail and specificity about the workflow. These states allow for finer control and tracking of workflow progress, and are available to Opus Solution Designers to configure.

#### **Business Object Example**





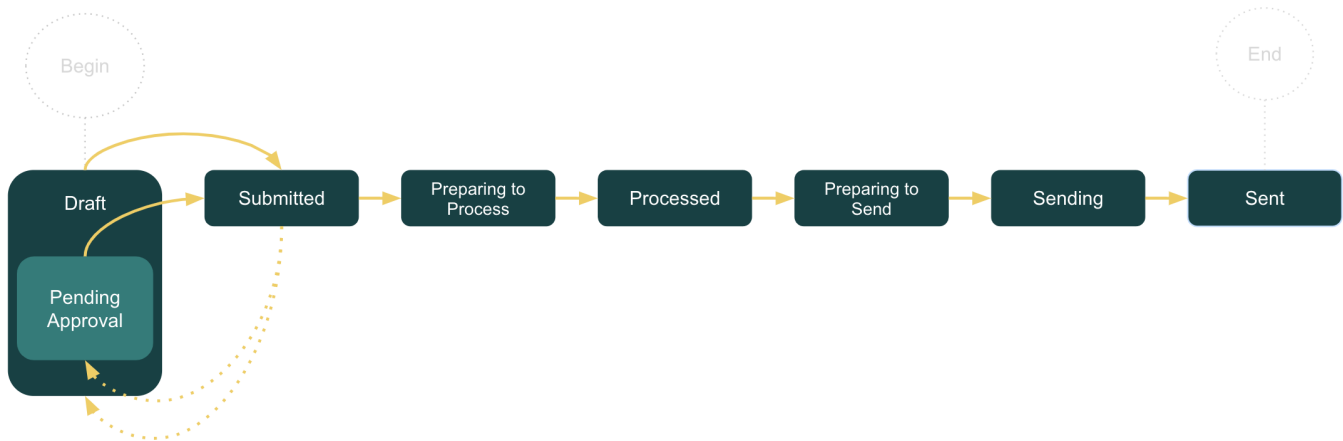
- To Do
- In Progress
  - **Drafting**
  - **In Review**
- Done

Keeping with the task example, you can see how an “In Review” substate was added to a task within the greater In Progress base state. For example, a Procurement Manager who has competing priorities may be tracking many of their “to dos” in SCWM's tasks. One of those tasks is preparing their budget for the upcoming quarter, which they will want their superior’s input on before putting it in the system. An “In Review” substate can be added to the In Progress base state to facilitate this process.

Substates have their own transitions that move between base states and substates. If substates are added, the base state where those substates reside will no longer be the state reflected in the flow (but merely a container for the substates). Therefore, you may need to add a substate that serves in place of the original base state. When the Drafting substate is added, it replaces In Progress,

which also illustrates that the substate name should be unique and different from the base state.

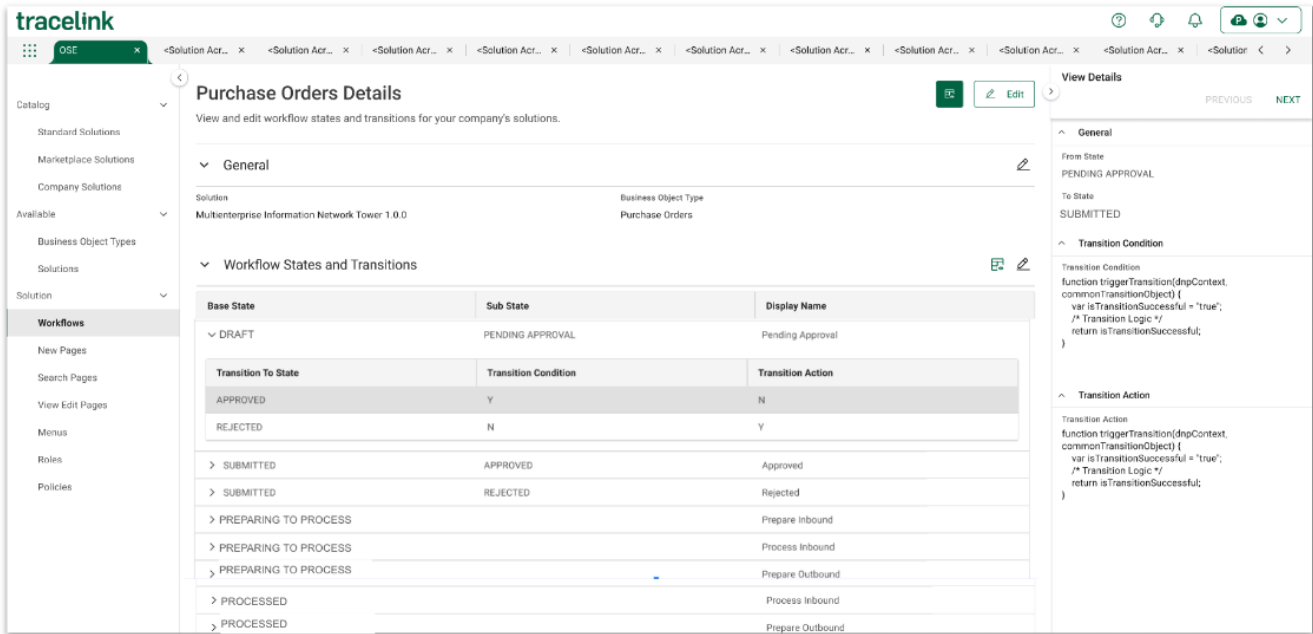
### BTO Example



- Draft
  - **Drafting**
  - **Pending Approval**
- Submitted
- Preparing to Process
- Processed
- Preparing to Send
- Sending
- Sent

In keeping with the Purchase Order example, a Procurement Manager may want to incorporate a review process on POs where the Net Price exceeds a maximum threshold so they can keep tabs on what they're spending to ensure their quarterly budget is not exceeded. Their Order Specialists can continue to prepare POs, but if the threshold is breached, then the workflow can facilitate this part of the business process by incorporating an additional substate within the Draft base state.

# Transitions



Workflow transitions move an object from its current workflow state to another state (whether a base state or a substate). The 2 elements that can be used to define transitions are transition conditions and post-transition actions. The management of these 2 elements ensures that objects move through their workflow in a controlled and predictable manner.

**Note:** In the UI, the transition is triggered with the MOVE TO button and Save operation.

Transition messages signal when a specific transition has occurred within a workflow, and broadcasts this event to other applications. These messages enable communication between applications, allowing them to react in real-time to changes, ensuring orchestrated processes are more efficient by leveraging automated workflows. These messages are crucial for enabling seamless communication between multiple applications, ensuring they are synchronized without manual intervention. By leveraging workflow subscriptions, applications can dynamically respond to these transitions, enhancing coordination, which we'll address in a later section.

## **Transition Conditions**

Transition conditions are the criteria that must be met for a transition to be considered valid, and therefore for the workflow to progress to the next defined state. If these conditions are not met, the transition is effectively treated as if it does not exist.

### **Business Object Example**

A typical task requires certain fields to be populated before it can be moved to a Done state. For example, a condition could be added where if an assignee is left blank (null), the task would stay in the In Progress base state (In Review substate), unable to move to a Done state.

### **BTO Example**

From our earlier example, if a PO requires a Net Price field to be populated before it can move to a Submitted state, then if that field is null, the PO cannot move to the Submitted state. Therefore, a condition for transitioning from Draft or Pending Approval to Submitted might be that the Net Price field must be present (i.e. not null).

## **Post-Transition Actions**

Post-transition actions are pieces of JavaScript code that execute after a transition has occurred (e.g. when a state change happens). These actions enable additional logic to be added by specifying what should happen when a transition is successful or unsuccessful (e.g. post-transition action errors, do not cancel the transition). This is a key mechanism that helps define specific behaviors or processes when the state transition occurs. These are often used to send notifications or trigger some other post-processing activity.

## **Workflow Quick Reference**

These are the key points from the earlier sections.

## Standard Workflows

- **Purpose** – Out-of-the-box functionality for an object type using base states and transitions.
- **Characteristics** – Simple representation of a business object and more robust representation of a BTO in its lifecycle.
- **Configuration** – There is no configuration on standard workflows; they are created by Opus Developers.

## Business Object Workflows

- **Purpose** – Flexible functionality where the out-of-the-box workflow for an object type is extended using base states and optionally substates, along with transitions.
- **Characteristics** – A more granular representation of an object's position in its greater lifecycle to meet unique business needs.
- **Configuration** – Configured by Solution Designers with a combination of base states and substates, or by using a standard workflow in a company solution.

## Base States

- **Purpose** – Represent major phases or milestones in a workflow.
- **Characteristics** – High-level, significant, and broad in scope.
- **Configuration** – Defined by Opus Developers at the initial setup of an object type workflow and usually changes frequently.

## Substates

- **Purpose** – Represents more granularity of specific business processes.
- **Characteristics** – Can provide additional detail within a base state, allowing for finer control.
- **Configuration** – Solution Designers can create and configure substates.

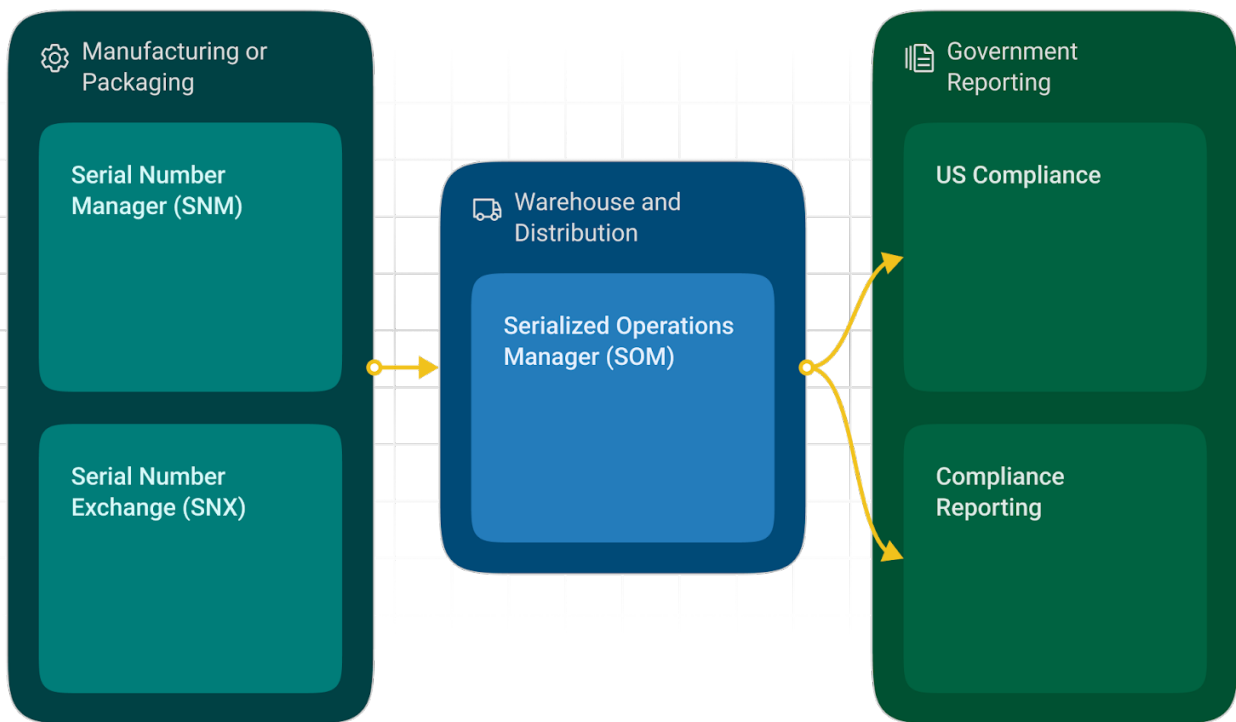
## Transitions

- **Purpose** – Enable the movement of objects from one workflow state to

another.

- **Characteristics** - Governed by transition conditions and can trigger post-transition actions. Can be automatic or manual.
- **Configuration** - Solution Designers can configure these constructs within a company solution.

## Workflow Subscription



A **workflow subscription** consists of predefined actions that trigger when an object transitions to a new state. These actions can be initiated by events outside of the solution. For example, if a SOM transaction transitions to a new state and a workflow subscription for compliance reporting is enabled, the system can automatically generate a government report and send it to the appropriate government entity based on the transaction's details.

In the Opus Platform, Opus Developers define these subscriptions for base states, which are not configurable by Solution Designers. Solution Designers can add

substates and define triggers based on those substates.

## **Example Workflow Subscription**

European Union (EU) Compliance Reports are generated based on certain events that occur in SOM. For example, if TraceLink processes a Batch Recall and that recall has a target market set to “EU FMD” (with a prerequisite that Traceability Required is set to true on product master data), then TraceLink generates a message to the EU Compliance application to create a Batch Recall report, which can subsequently be sent to the EU Hub to fulfill compliance requirements.

## **Benefits of Workflow Subscriptions**

1. **Real-Time Updates** – Subscriptions ensure that all relevant parties and entities are immediately informed of important events, enabling prompt responses and actions.
2. **Automation** – Workflow subscriptions reduce manual effort, minimize errors, and improve efficiency.
3. **Coordination** – Subscriptions help coordinate activities across different systems and teams, ensuring that all parts of the business process are aligned and working together.

## **Key Characteristics of Workflow Subscriptions**

1. **Event-Driven** – Workflow subscriptions are triggered by specific events within a workflow, such as state transitions, the completion of a particular step, or the occurrence of an error.
2. **Configurable** – Workflow subscriptions can be configured to trigger automated actions (e.g. calling an API or updating a database).
3. **Integration** – Workflow subscriptions facilitate integration between different products and business systems, ensuring seamless coordination and data flow.

**Related Content**



**TraceLink's Opus Platform**

TraceLink's Opus is a unique platform as a service (PaaS) that integrates a metadata-driven, no-code solution with the world's only end-to-end global life sciences supply chain network, delivering superior business outcomes by optimizing complex multienterprise processes to enhance operational efficiency, mitigate risks, enable informed decision-making, and strengthen overall supply chain agility and resilience.

**[View More](#)**





### **Transforming Supply Chain Orchestrations With Opus Solution Environment (OSE)**

The Opus Solution Environment (OSE) empowers organizations to rapidly adapt and innovate by allowing Opus Solution Designers, regardless of coding skills, to create robust solutions without code, while also supporting advanced configurations with JavaScript for greater flexibility.

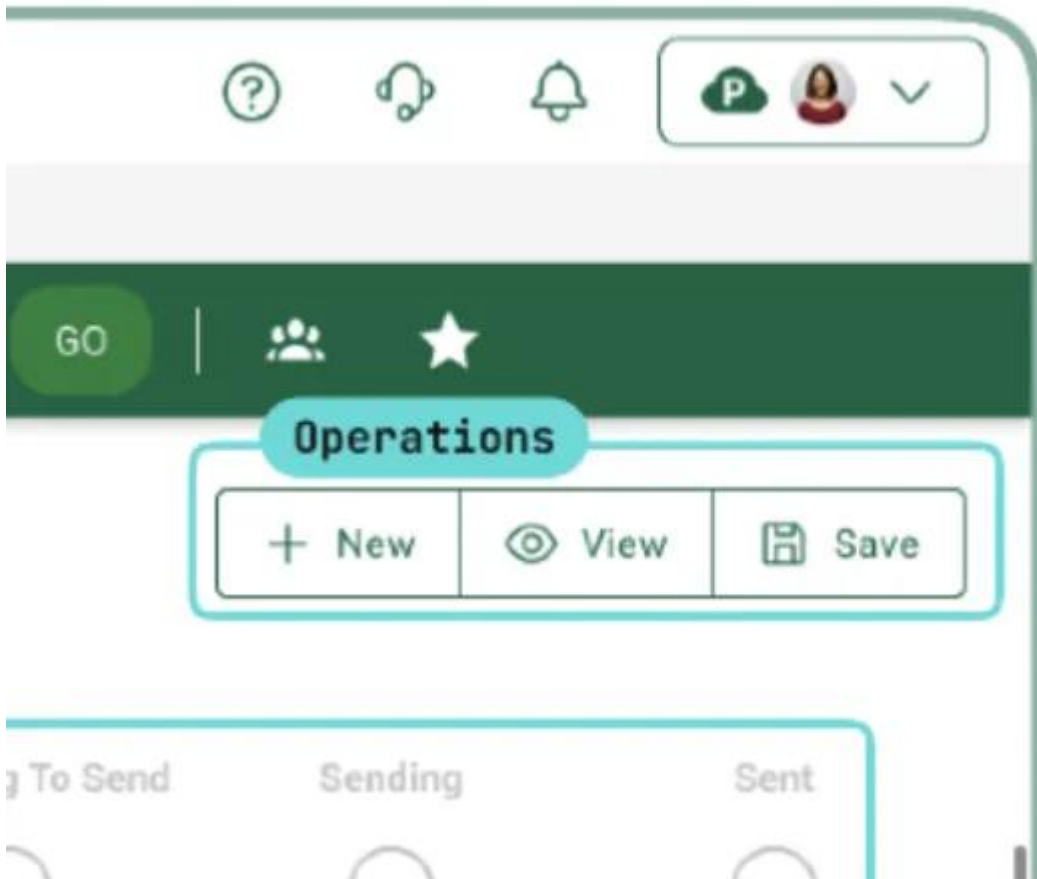
**[View More](#)**



### **The Opus Platform's Metadata-Driven, No-Code Solutions**

Opus Solution Designers can create solutions on the Opus Platform without being data modelers or developers, but understanding core modeling constructs and the benefits of the Opus approach is beneficial before using the Opus Solution Environment (OSE).

**[View More](#)**



### **Optimizing Solution Delivery With Predefined Object Operations in Opus**

Object operations are central to Opus's no-code, metadata-driven framework, enabling user interaction with business objects through predefined actions, with automatic generation of these operations from metadata configurations that enhance consistency and accelerate solution delivery.

**[View More](#)**



### **Get Started Designing on the Opus Platform With Opus Anthem**

Anthem is TraceLink's Design System that establishes consistent design conventions across the Opus Platform, allowing designers to quickly create solutions that meet various business and user needs.

**[View More](#)**



### **Opus Solution Environment**

A no-code, drag-and-drop environment that enables business users to tailor the workflows and user experiences of TracLink solutions to the unique requirements of their supply chain without the need for specialized technical skills or extensive coding knowledge.

**[View More](#)**