Acumatica
The Cloud ERP

W150 Workflows

Nicholas Bova

Services Developer

Timing and Agenda

September 18, 2023 -10:00 AM -11:30 AM

Day 1

Lesson 1.1: Getting Started with Workflows

Lesson 2.1: Creating Workflows

Lesson 2.2: Configuring Fields and Dialog Boxes

Lesson 2.3: Configuring Actions

September 19, 2023 -10:00 AM -11:30 AM

Day 2

Lesson 2.4: Configuring Conditions and Transitions

Lesson 2.5: Testing the Customization

Lesson 3.1: Creating an Inherited Workflow

Lesson 3.2: Organizing the Workflow by Using Visual Editor

Lesson 3.3: Modifying the Actions by Using Visual Editor

Lesson 3.4: Finalizing Customization of the Workflow

Timing and Agenda

September 20, 2023 -10:00 AM -11:30 AM

Day 3

Lesson 3.5: Modifying a Workflow with a Composite State

Lesson 4.1: Creating an Inherited Workflow with a Workflow-Identifying Field

Lesson 4.2: Implementing Automated Transitions

Lesson 4.3: Modifying System Actions and Testing the Customization Project

Lesson 5.1: Upgrading Workflows

Part 1: Getting Started with Workflows

Part 2: Creating a Workflow

Part 3: Working with Inherited Workflows

Part 4: Customizing a Workflow with a Workflow-Identifying Field

Part 5: Upgrading Workflows

Joe Gibbs Racing
Acumatica Partner

Day 1

Company Story and Customization Description

You will perform the exercises of this course acting as an employee of a partner of Acumatica that develops business solutions for companies. You are going to customize some of the Acumatica ERP forms for HardwareViewpoint, a midsize hardware manufacturer that has decided to replace its aging ERP system with Acumatica ERP.

In HardwareViewpoint, the flow of documents is based on certain procedures, and you want to make modifications to the available workflows in Acumatica ERP so that the workflows better suit the company's needs. Specifically, you want to do the following:

- Create a workflow for the Task (CR306020) form, which does not have any predefined workflows in an out-of-the-box system.
- Support the assignment of opportunities to salespersons. To provide this support, you will customize the Opportunities (CR30400) form to add a new status, Assigned, after the New status in the workflow. The Assigned status reflects that the opportunity has been assigned to a salesperson, who must then accept or reject the opportunity. If the opportunity is accepted, the opportunity status becomes Open; if it is rejected, the status changes back to New.
- Set up the system to change the status of a sales order to On Hold automatically if the OrderTotal value is $800 or more. To do this, you will customize the Sales Orders (SO30100) form.
- For an inherited worklfow that has been developed for a previous version of Acumatica ERP, upgrade the customization project that contains this inherited workflow because the predefined workflow of a form has been modified since the previous version.

Lesson 1.1: Getting Started with Workflows

Learning Objectives

In this lesson, you will learn how to do the following:

- What workflows are and how you can use them.

Figure: Steps in a transition for an action

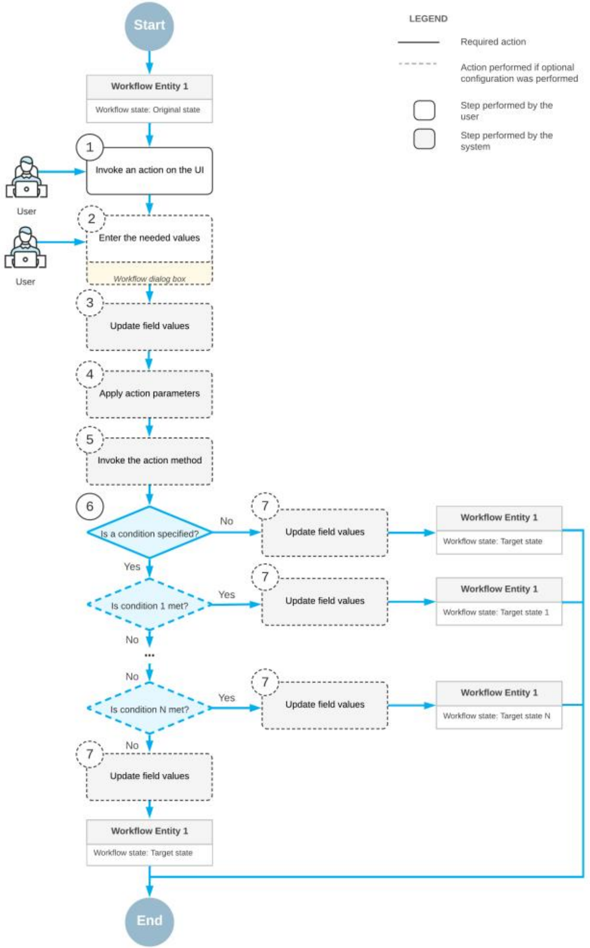
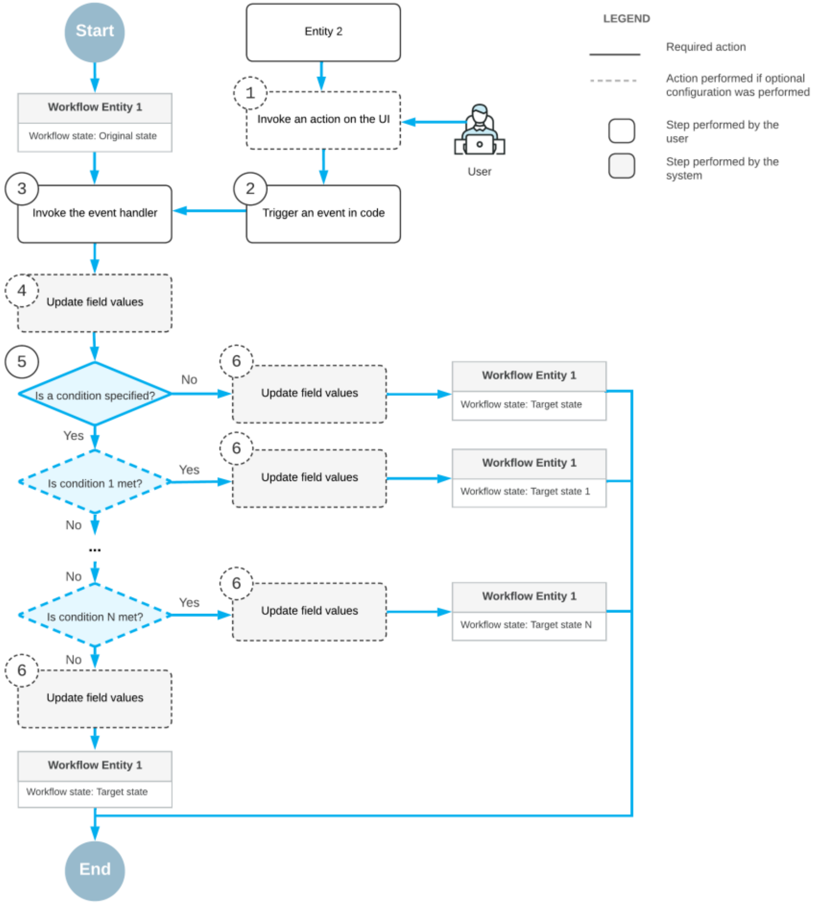


Figure: Steps in a transition for an event handler



Lesson 2.1: Creating Workflows

Learning Objectives

In this lesson, you will learn how to do the following:

- To use the Workflow (Tree View) page, and how to create a workflow from scratch.

Figure: The Workflows page

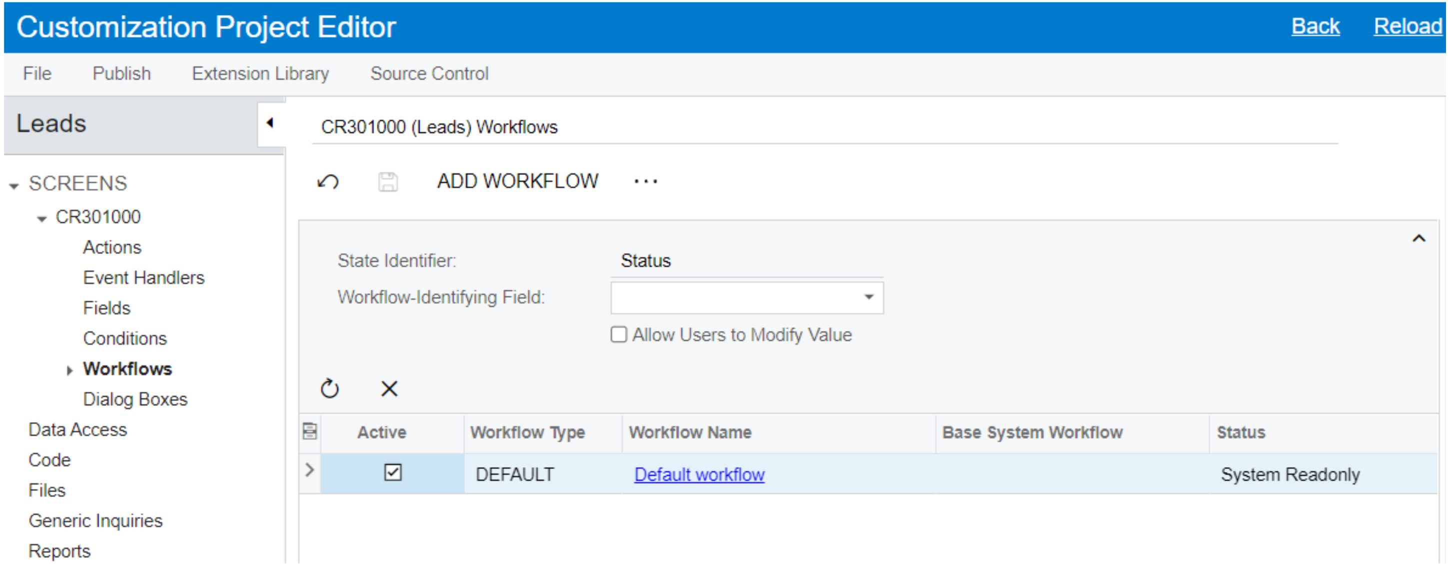


Figure: Workflow page (tree view)

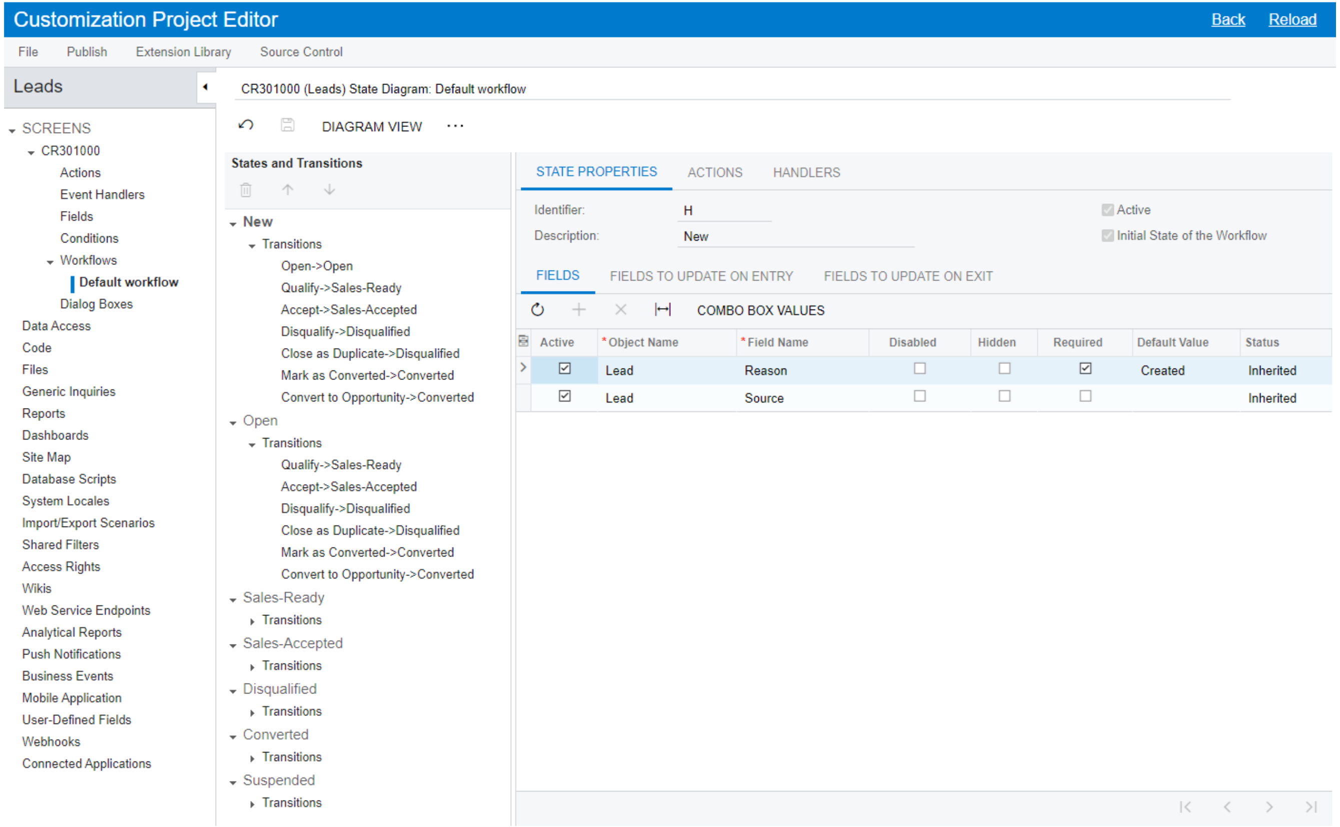
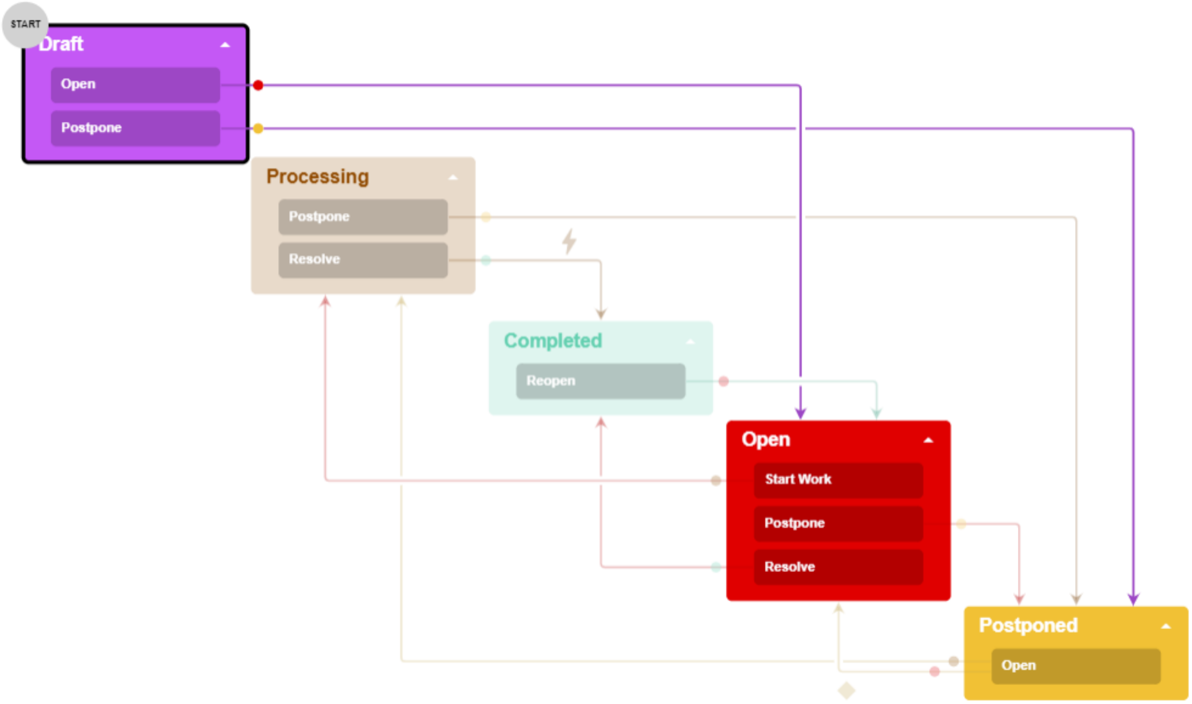


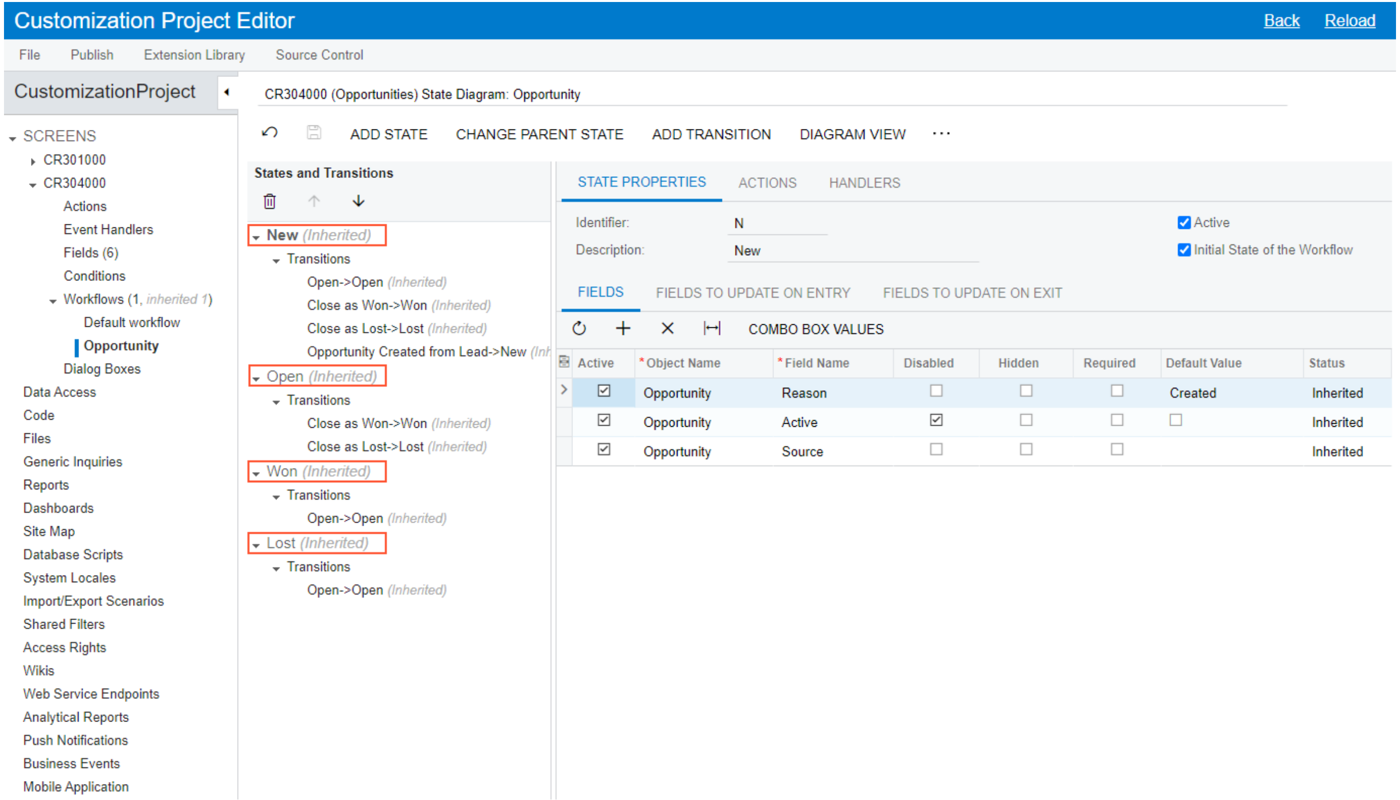
Figure: The planned task workflow



Workflow Creation: To Add a Workflow

Suppose that you are a technical specialist that develops business solutions for the HardwareViewpoint company. You need to create a workflow for the Task (CR306020) form, which is planned as specified in Workflow Creation: Planning a Workflow for a Form.

Figure: States on the State and Transitions pane



Workflow Creation: To Add States

Acting as the technical specialist, you need to add predefined states and a new state to the workflow you have defined for the Task (CR306020) form.

Figure: The Add State dialog box

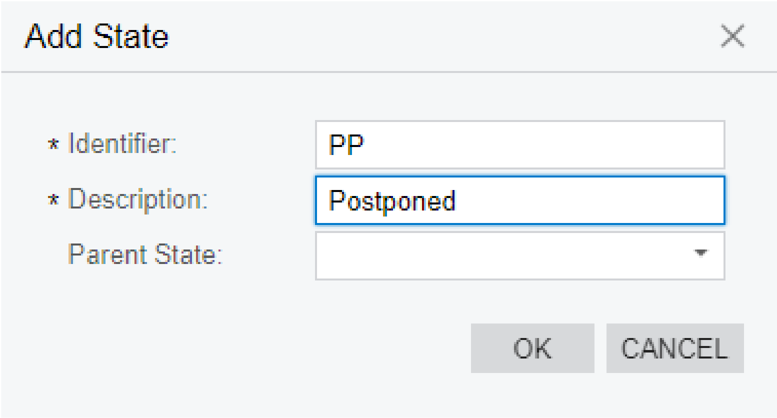
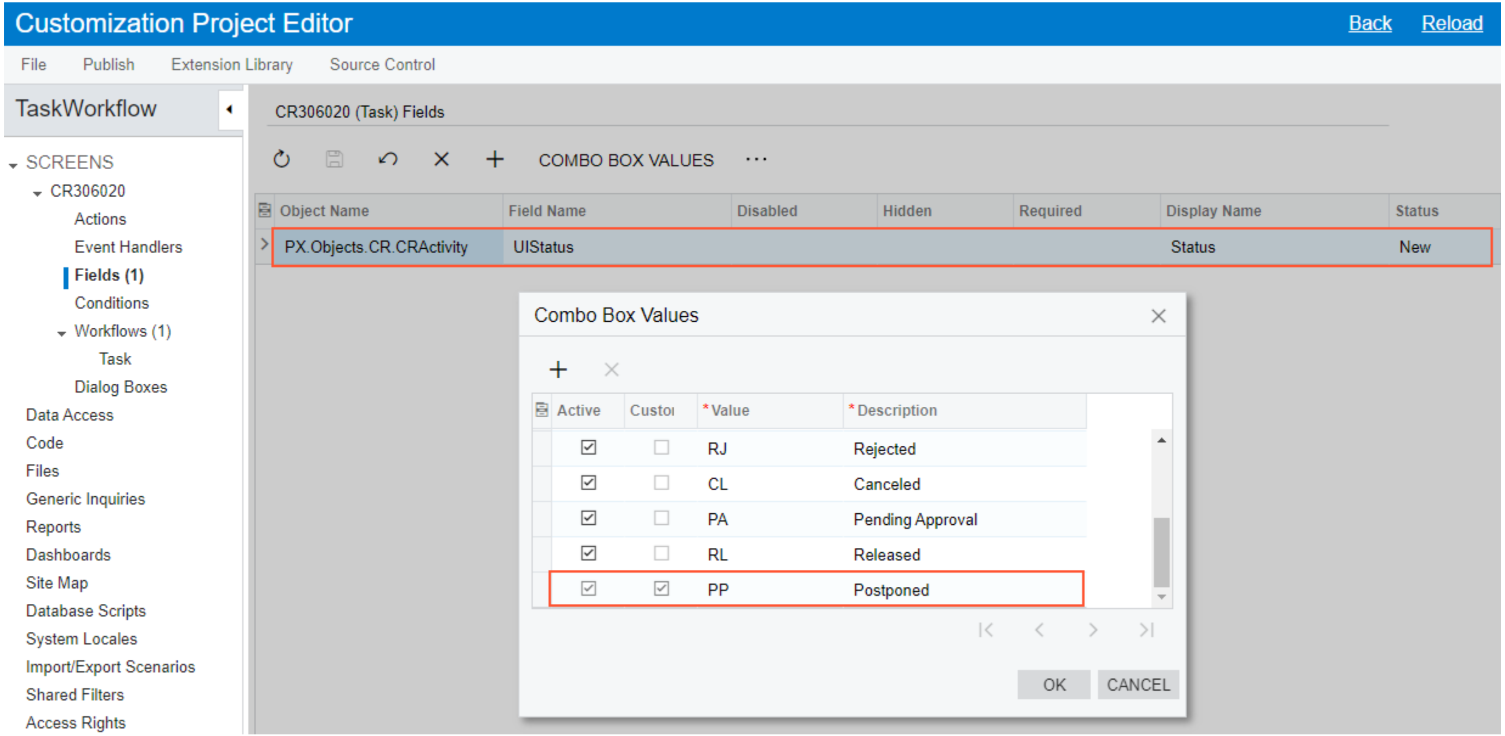


Figure: The Fields page with the added field



Acumatica
The Cloud ERP

Lesson 2.2: Configuring Fields and Dialog Boxes

Learning Objectives

In this lesson, you will learn how to do the following:

- To modify field settings and create dialog boxes.

Figure: The CR301000 (Leads) Fields page

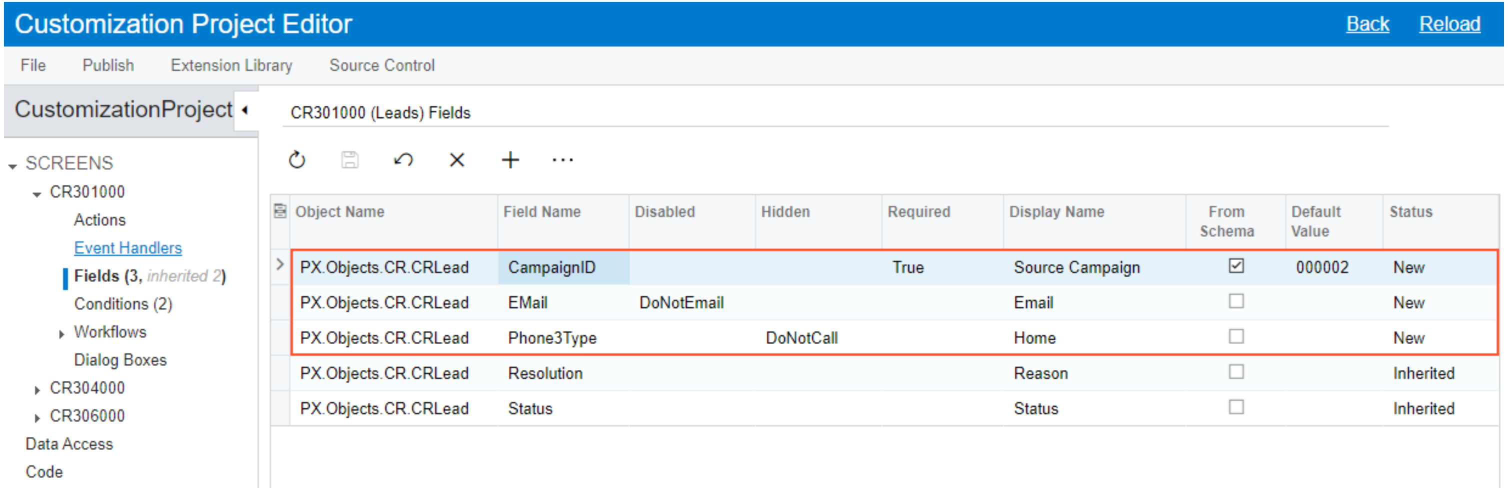


Figure: The CR306000 (Cases) Dialog Boxes page

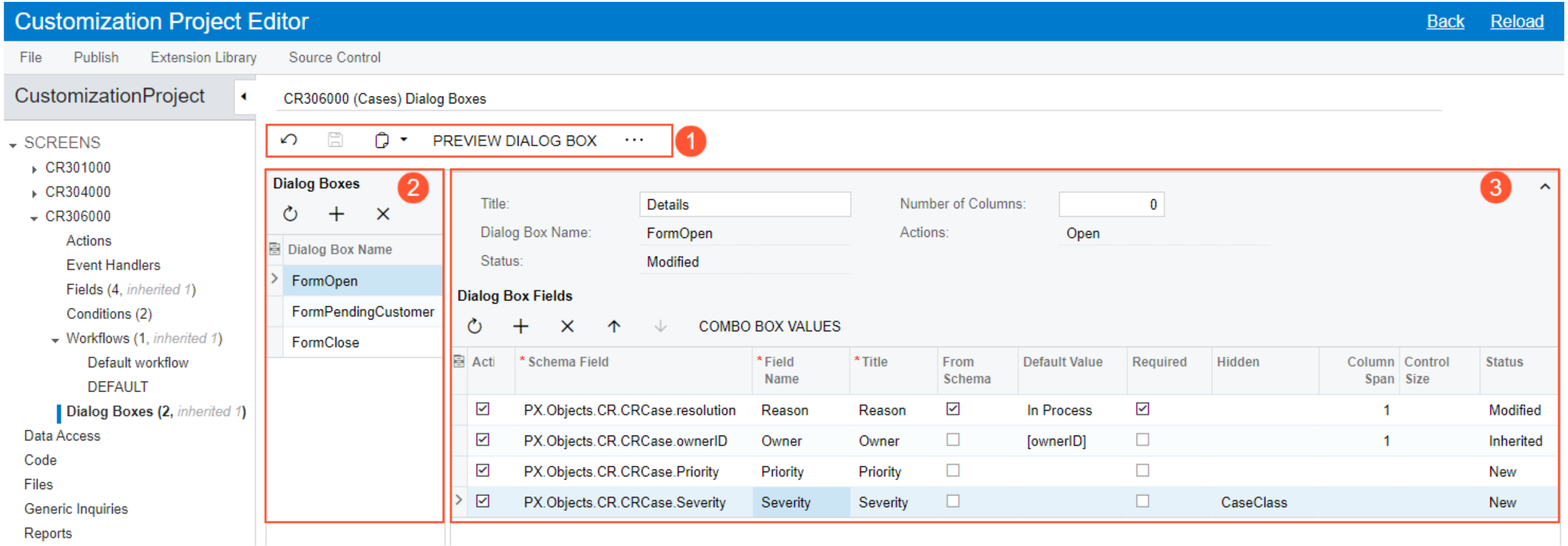
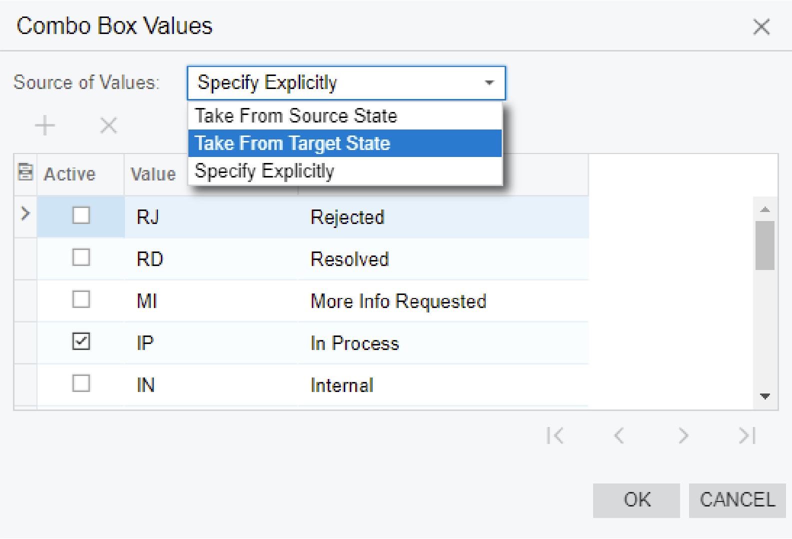


Figure: The Source of Values box of the Combo Box Values dialog box



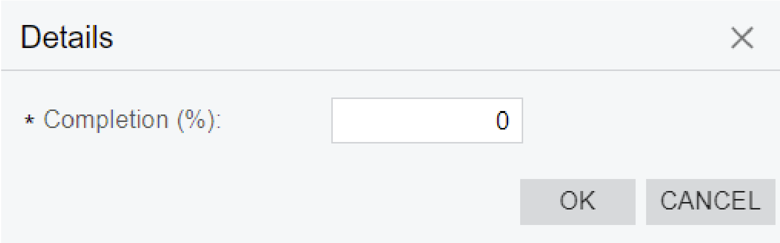
Workflow Elements: To Modify Field Settings

Acting as the technical specialist, you need to modify the field settings for the workflow states you have added in Workflow Creation: To Add States. With the modified settings, when a task is created, the value in theStart Date box will be the current date, and the Completion (%) box will be unavailable for editing.

Workflow Elements: To Add a Dialog Box

According to the planned workflow of the Task (CR306020) form, when a user reopens a completed task on the form, the system needs to display a dialog box. In this dialog box, the user must specify the percent of task completion. Acting as the technical specialist, you will create this dialog box.

Figure: The Details dialog box for the Reopen action



Lesson 2.3: Configuring Actions

Learning Objectives

In this lesson, you will learn how to do the following:

- Hide unneeded actions on the form
- Add new actions to the workflow
- Modify the added actions
- Add categories of the More menu
- Organize the actions into categories on the More menu

Figure: Modified action

SO301000 (Sales Orders) Actions

REORDER ACTIONS ···

Action Name	Display Name	Action Type	Disabled	Hidden	Dialog Box	Processing Screen	Category	Status
printSalesOrder	Print Sales Order	Graph Action				SO502000	Printing and Em...	Inherited
processExpired...	Process Expired ...	Graph Action				SO501000	Processing Cate...	Inherited
putOnHold	Hold	Graph Action					Processing Cate...	Modified
putOnHoldAuto	putOnHoldAuto	Workflow		True				New
quickProcess	Quick Process	Graph Action		NOT(AllowQuick...			Processing Cate...	Inherited
ReassignApproval	Reassign	Graph Action		True			Action	Inherited
recalcExternalTax	Recalculate Exte...	Graph Action					Other Category	Inherited
recalculateDisco...	Recalculate Prices	Graph Action					Other Category	Inherited
reject	Reject	Graph Action		True			Action	Inherited
releaseFromCre...	Remove Credit ...	Graph Action				SO501000	Approval Category	Inherited
releaseFromHold	Remove Hold	Graph Action					Processing Cate...	Modified
removeHoldTotal...	Total Less Than ...	Workflow		True			Actions	New
removeRiskHold	Remove Risk Hold	Graph Action					Approval Category	Inherited
reopenOrder	Reopen Order	Graph Action					Processing Cate...	Inherited
riskHold	Risk Hold	Graph Action					Approval Category	Inherited
ShowCustomerD...	Customer Details	Navigation: Side ...		IsTransfer				Inherited
ShowInvoicesAn...	Invoices and Me...	Navigation: Side ...		NOT(CanBeInv...				Inherited
validateAddresses	Validate Addresses	Graph Action					Other Category	Inherited

Action Configuration: To Hide Unneeded Actions

In your workflow, you do not need the Cancel, Complete, and Complete & Follow-Up buttons or the corresponding commands in the More menu on the Task (CR306020) form. Acting as the technical specialist, you are going to hide these buttons and menu commands on the form.

Action Configuration: To Add a Category to the More Menu

Suppose that you want to add all the actions in the new workflow for the Task (CR306020) form to the Processing category of the More menu. However, this form does not have any categories specified for the menu, and the More menu is not displayed. Acting as the technical specialist, you need to add a category to the More menu of this form.

Action Configuration: To Create Workflow Actions and Add Them to the Workflow States

Acting as a technical specialist, you need to create new Open, Postpone, StartWork, and Resolve actions for the Task (CR306020) form. These actions will be workflow actions, which move a task from one state of the workflow of the form to another state.

You need to add the new actions to the following states:

- The Draft state should contain the Open and Postpone actions.
- The Processing state should contain the Resolve and Postpone actions.
- The Open state should contain the StartWork, Resolve, and Postpone actions.
- The Postponed state should contain the Open action.

Action Configuration: To Create a Workflow Action That Displays a Dialog Box

When a task on the Task (CR306020) form has the Completed state, only one action should be available: the Reopen action. When a user clicks the Reopen button or command, you want the system to display the Reopen dialog box, in which the user must specify the percent of the task completion.

Acting as the technical specialist, you need to create an action and add it to the Completed state.

Day 2

Lesson 2.4: Configuring Conditions and Transitions

Learning Objectives

In this lesson, you will learn how to do the following:

- Create conditions that will be used to modify actions. You will also learn how

to add transitions between the states of the workflow.

Figure: The Conditions page

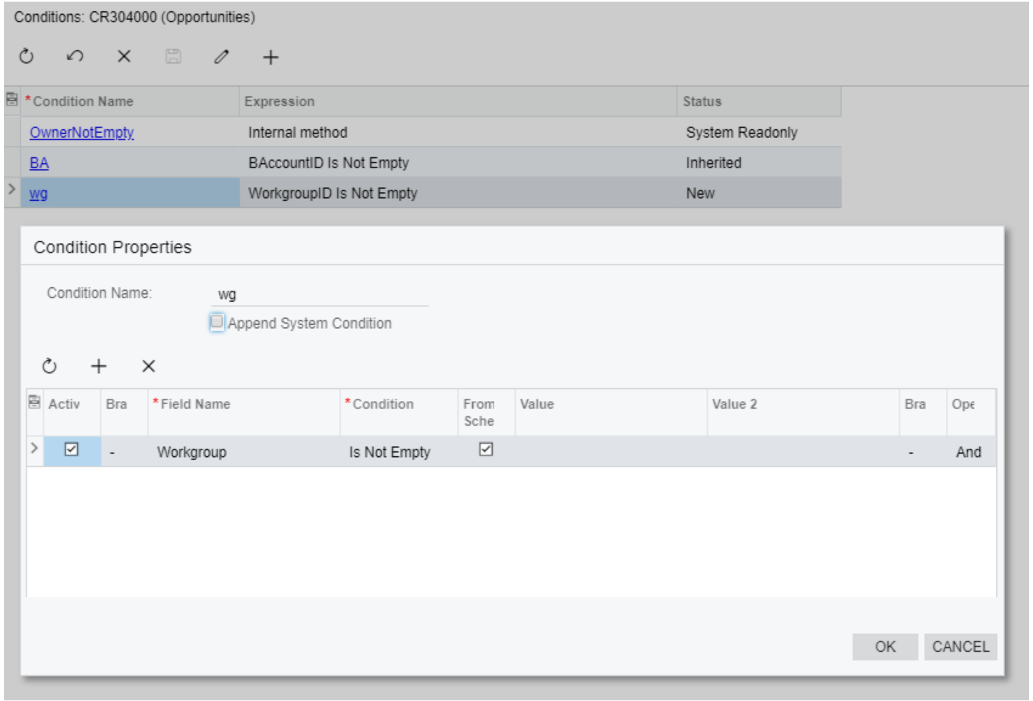
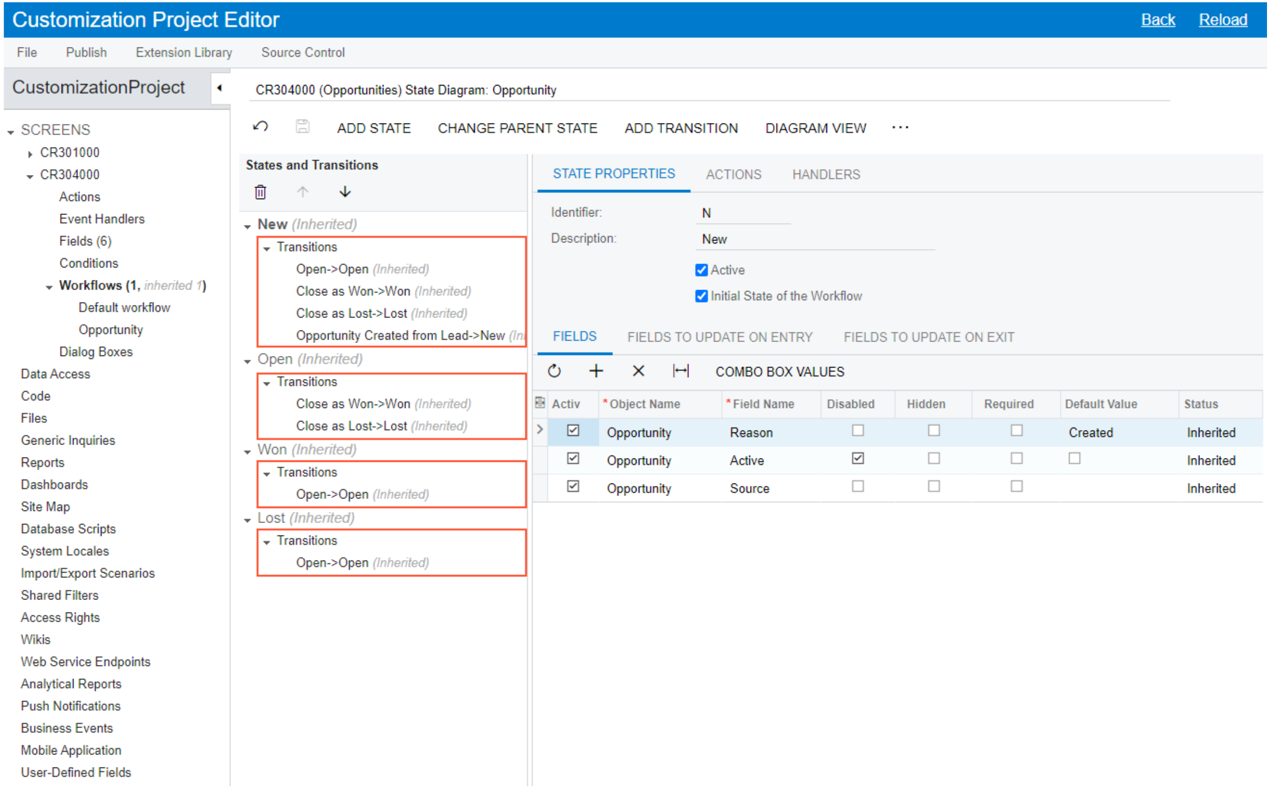


Figure: The State and Transitions pane



Conditions and Transitions: To Add Conditions

In your workflow for tasks created on the Task (CR306020) form, you need the system to perform the Resolve action automatically when the Completion (%) is set to 100, which reflects that the task is complete. You also want the Open action to trigger transitions to different states depending on the percent of the task completion. In this activity, you will start this process by creating the conditions.

Conditions and Transitions: To Add Transitions

You can use the same action to trigger transitions from one state to different states. To do so, you need to use conditions for these transitions. If a condition is fulfilled, the system performs the transition that is first in the list of transitions from the current state. (This list is displayed on theStates and Transitions pane of the Workflow (Tree View) page.) If the condition is not fulfilled, the system switches to the next transition in the list and checks its condition.

In the Postponed state, you will use the same action (the Open action) to trigger transitions to the Open state or to the Processing state, depending on the percent of the task completion as follows:

• If the value in the Completion (%) box is 0, the target state should be Open.

• If the value in the Completion (%) box is not 0, the target state should be Processing.

To implement these transitions, you will use the NotStarted and Completed conditions, which you have created in Conditions and Transitions: To Add Conditions.

Lesson 2.5: Testing the Customization

Learning Objectives

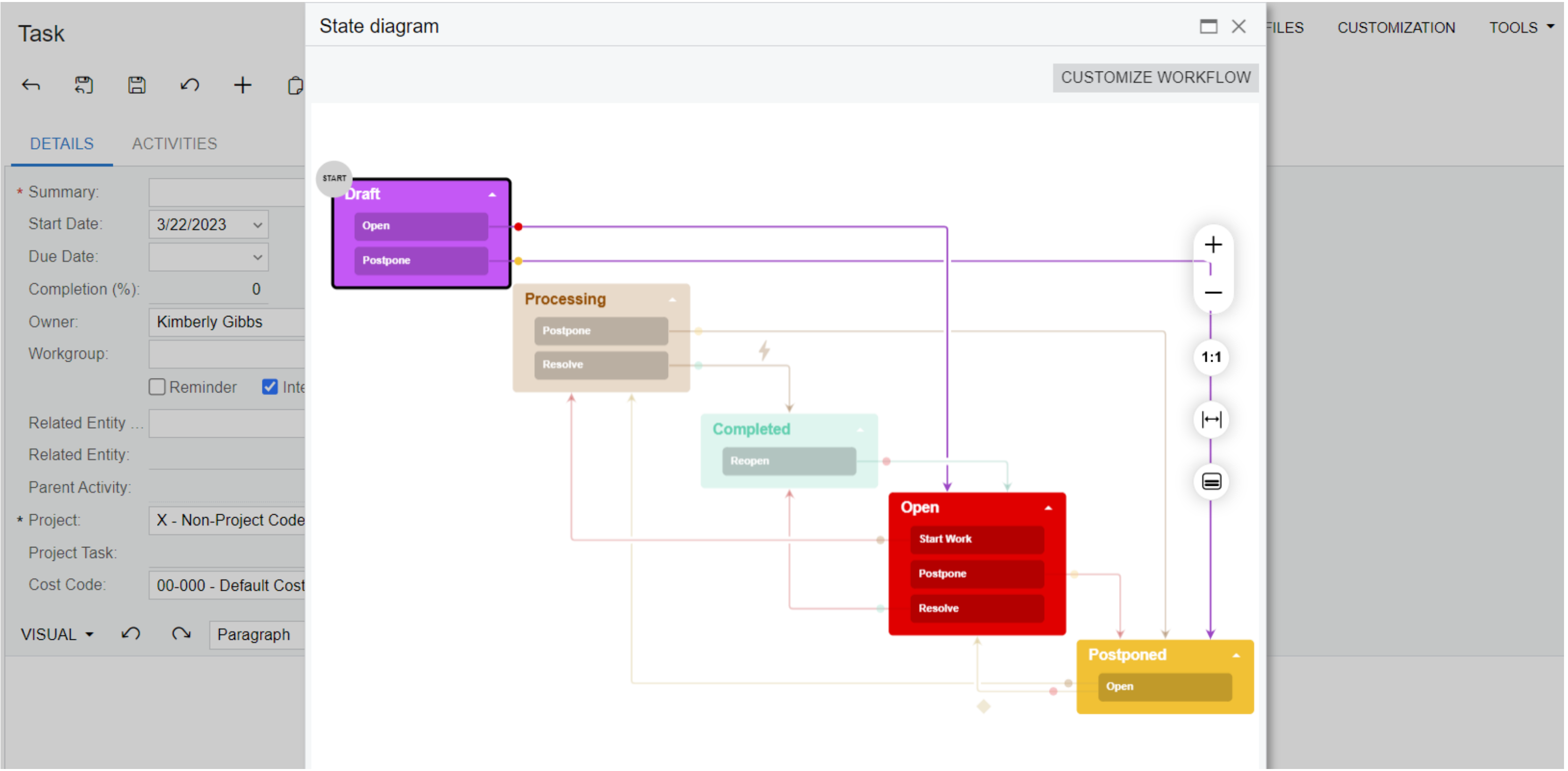
In this lesson, you will learn how to do the following:

- Test your customization project.

Testing of the Customization: To Test a Custom Workflow

Acting as the technical specialist, you need to publish your customization project and then test the changes on the Task (CR306020) form and make sure that the new workflow works as expected.

Figure: The State Diagram dialog box



Lesson 3.1: Creating an Inherited Workflow

Learning Objectives

In this lesson, you will learn how to do the following:

- Work with inherited workflows, which are based on predefined workflows for the applicable forms.

Figure: The tree view of the Workflow page with the predefined workflow

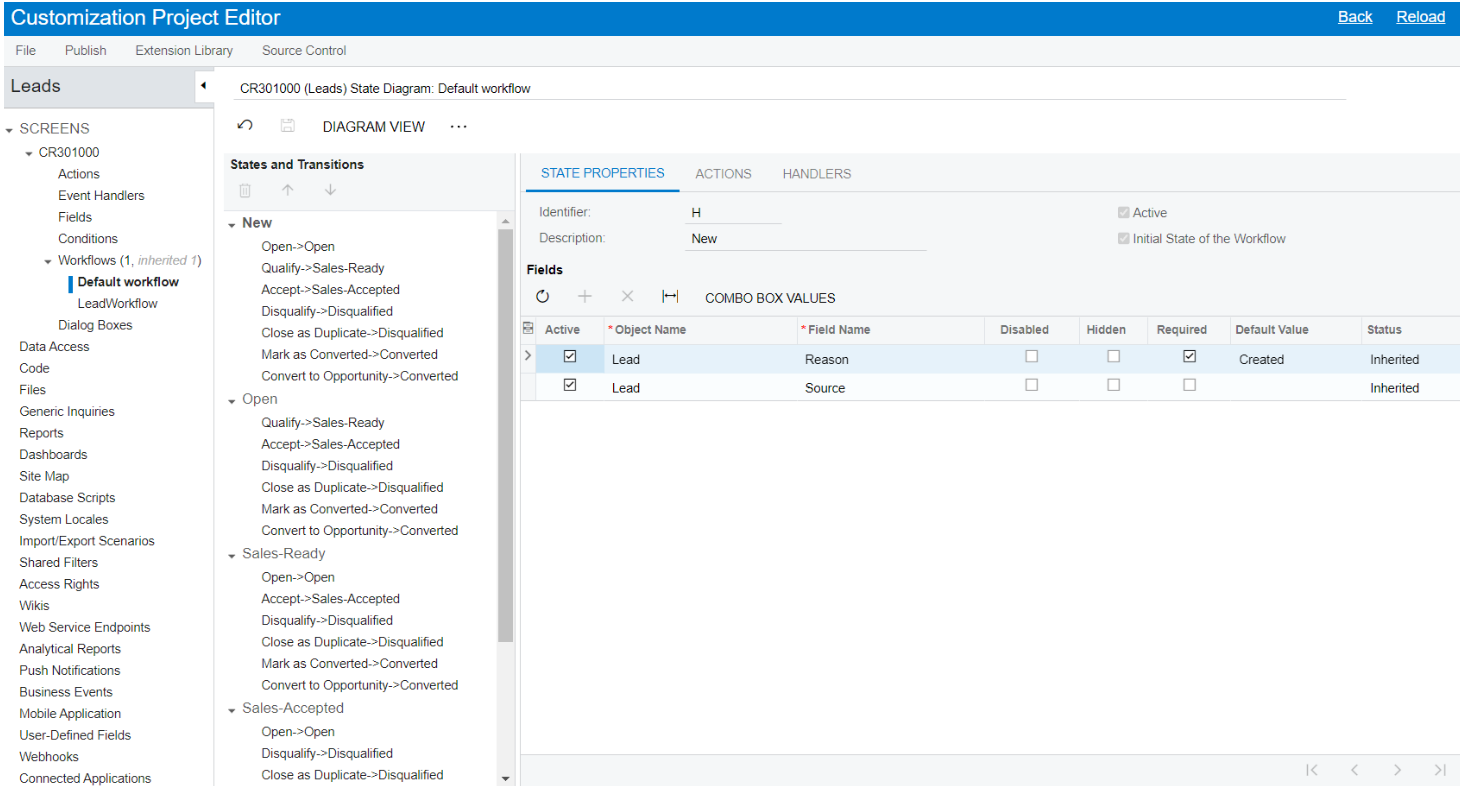


Figure: The predefined workflow for opportunities

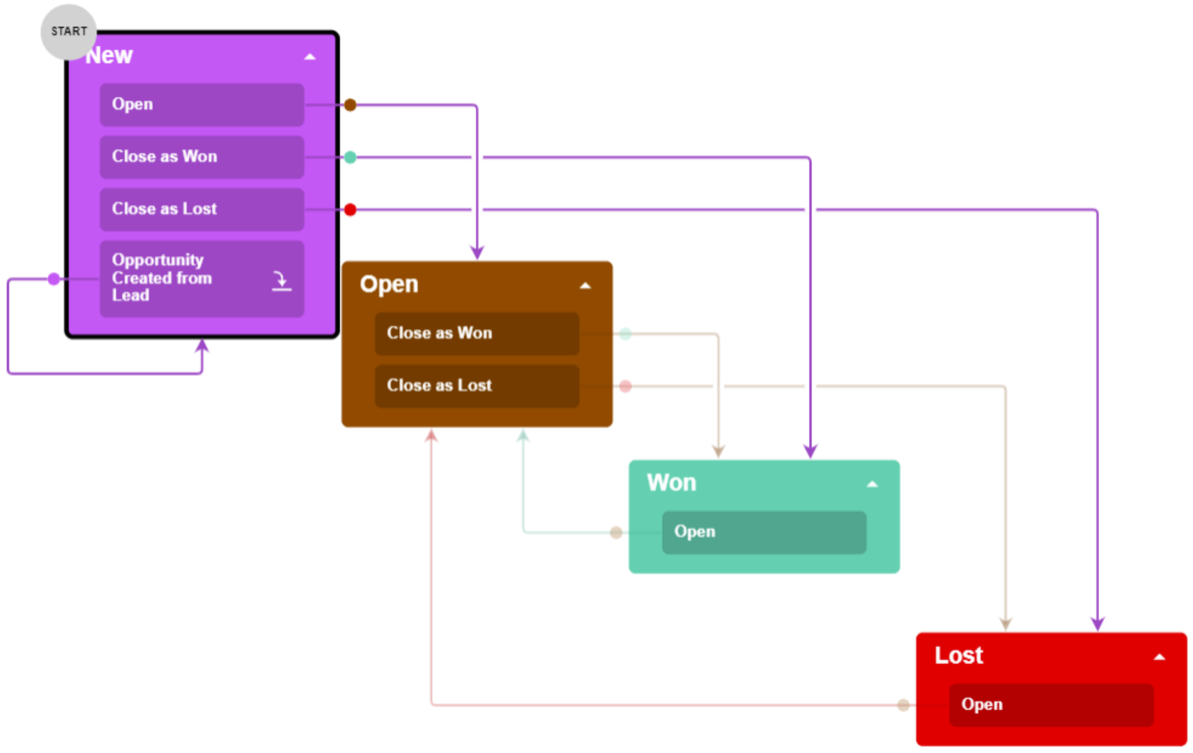
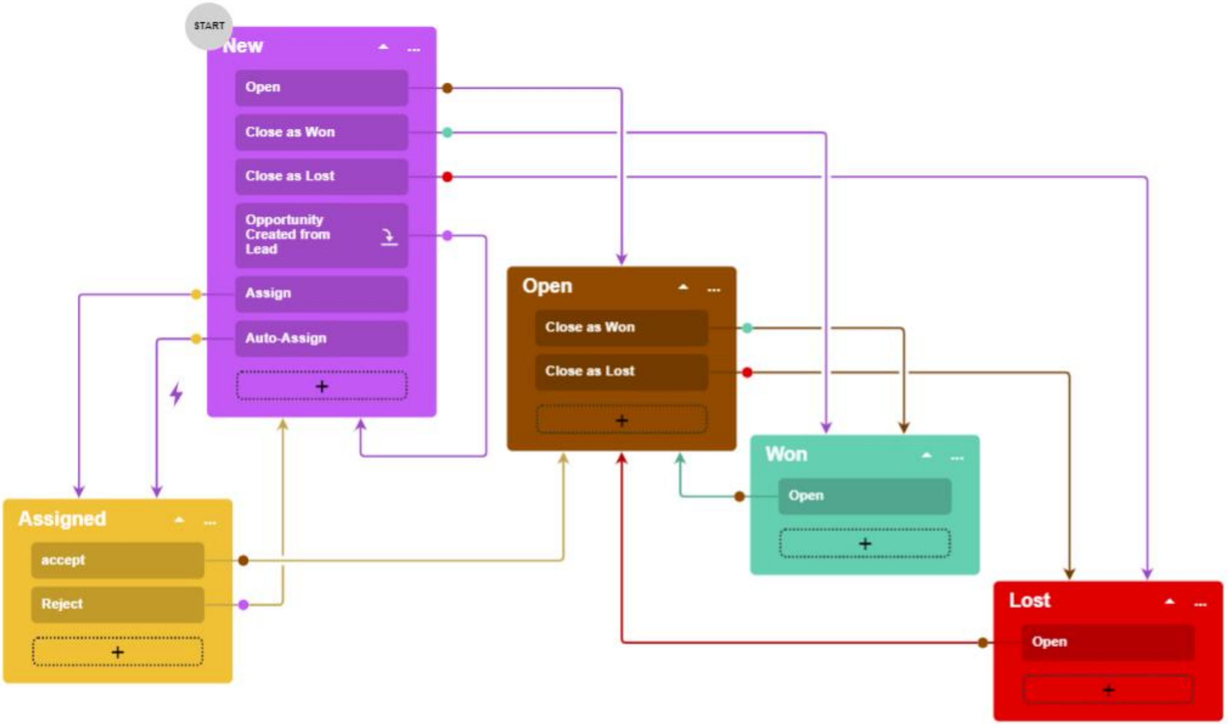


Figure: The modified workflow for opportunities



Inherited Workflows: To Create an Inherited Workflow

Acting as the technical specialist, you need to create a customized workflow for the Opportunities (CR304000) form.

Lesson 3.2: Organizing the Workflow by Using Visual Editor

Learning Objectives

In this lesson, you will learn how to do the following:

- Use of the Workflow (Diagram View) to customize workflows.

Figure: The parts of the diagram view

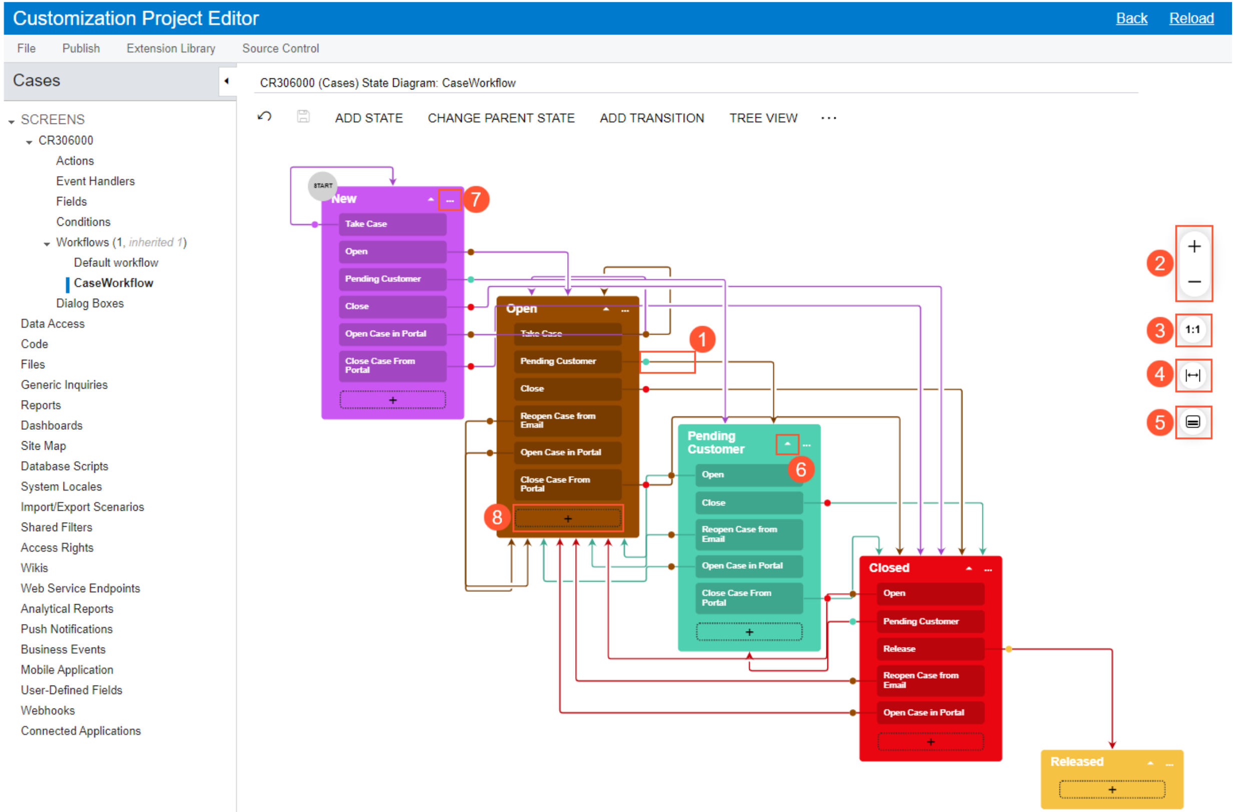


Figure: Parts of the transitions

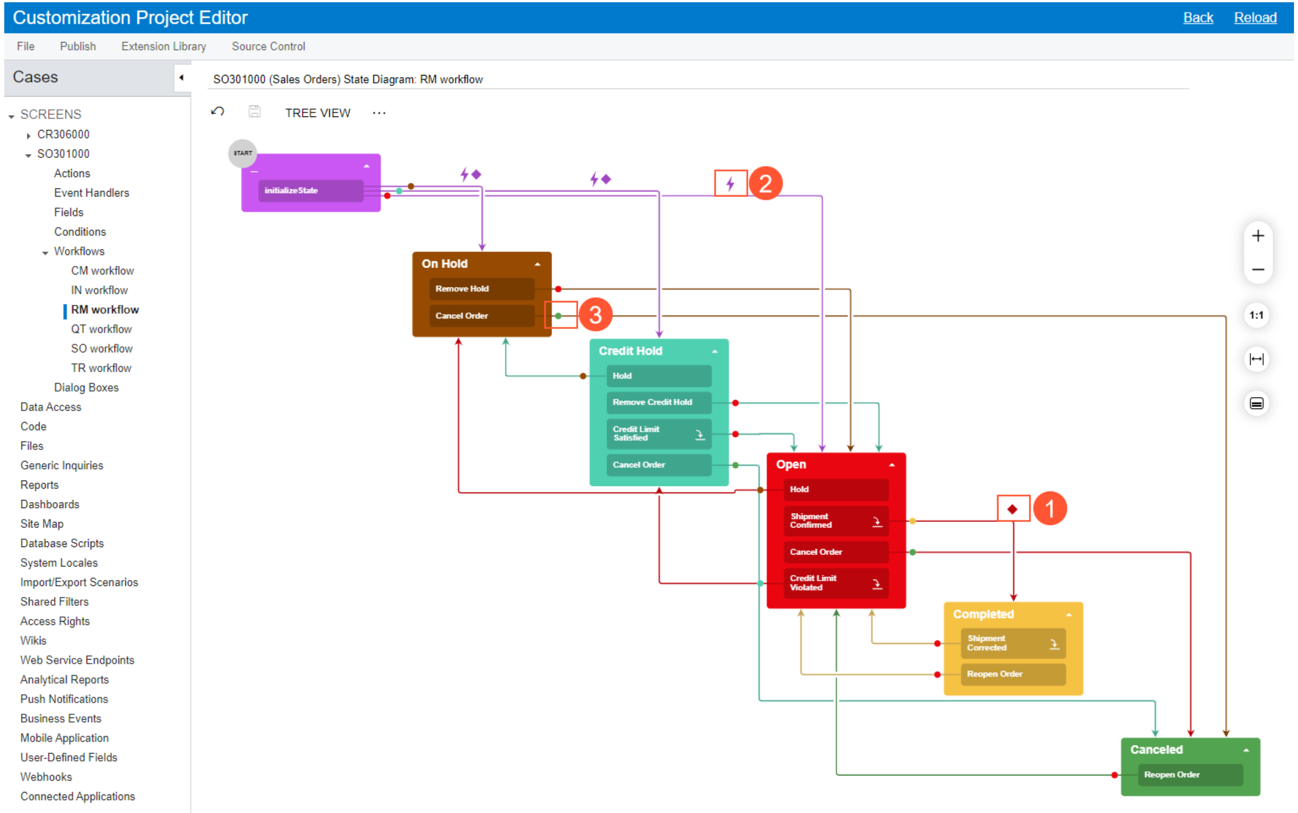


Diagram View: To Add a New State

Suppose that in your customization efforts on the Opportunities (CR304000) form, you need to incorporate a process of assigning a new opportunity before giving it the Open status. You need a new status, Assigned, for opportunities. Acting as the technical specialist, you need to add a new state to the opportunity workflow for the Opportunities form. This state will correspond to the Assigned status of an opportunity created on this form.

Diagram View: To Add New Actions and Transitions

Suppose that in your customization efforts on the Opportunities form, you need to incorporate a process of assigning a new opportunity before giving it the Open status. A user should be able to do the following:

- Change the status from New to Assigned by clicking Assign on the form toolbar.
- Change the status from Assigned to Open by clicking Accept.
- Change the status from Assigned to New by clicking Reject.

Acting as the technical specialist, you need to create the actions and specify the corresponding transitions.

Diagram View: To Add the Auto-Run Action

If the Owner box on the Opportunities (CR304000) form is filled and the opportunity has the New status, the status of the opportunity should be automatically changed to Assigned. Acting as the technical specialist, you need to add another transition from the New state to the Assigned state for the workflow used by the form. The transition should be triggered automatically if the Owner box is not empty.

Figure: The workflow diagram with the added state and transitions

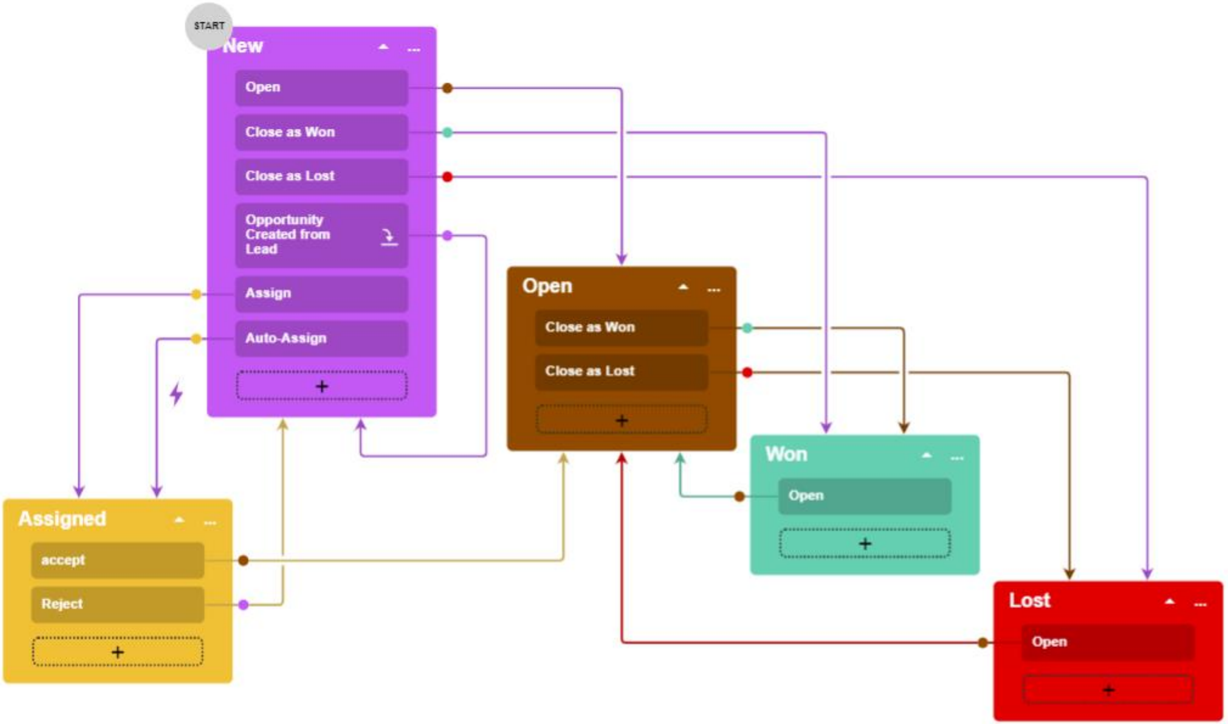


Diagram View: To Specify Combo Box Values for States

Acting as the technical specialist, you need to add to the workflow of the Opportunities (CR304000) form a new field for the Reason box and add new combo box values for this field.

Diagram View: To Add Dialog Boxes

In the HardwareViewpoint company, a salesperson should add additional information when performs the following actions with an opportunity on the Opportunities (CR304000) form of Acumatica ERP:

- While accepting the opportunity, specify the reason and the stage of the opportunity
- While rejecting the opportunity, select the reason and provide additional details for it
- While assigning the opportunity, select the owner of the opportunity

Acting as the technical specialist, you need to add dialog boxes that prompt users for this information when they click the corresponding commands or buttons to invoke the Accept, Reject, and Assign actions.

Lesson 3.3: Modifying the Actions by Using Visual Editor

Learning Objectives

In this lesson, you will learn how to do the following:

- Use the Workflow (Diagram View) page to modify the workflow

Figure: The Details dialog box for the Accept action

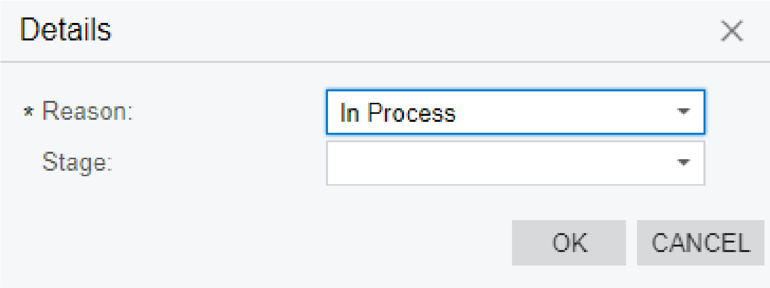


Diagram View: To Modify the Added Actions

In the HardwareViewpoint company, a salesperson should add additional information when performs the following actions with an opportunity on the Opportunities (CR304000) form of Acumatica ERP:

- While accepting the opportunity, specify the reason and the stage of the opportunity
- While rejecting the opportunity, select the reason and provide additional details for it
- While assigning the opportunity, select the owner of the opportunity

Acting as the technical specialist, you need to add the dialog boxes that prompt users for this information to the Accept, Reject, and Assign actions and modify the actions so that they use the information specified in the dialog boxes. You also need to display the actions on the Update Opportunities (CR503120) form for mass-processing of opportunities. You need to make the buttons that correspond to the Assign and Accept actions be displayed on the form toolbar.

Figure: The Details dialog box for the Reject action

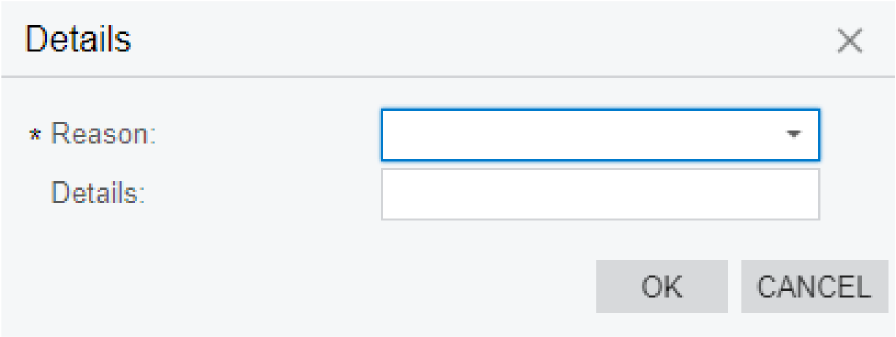
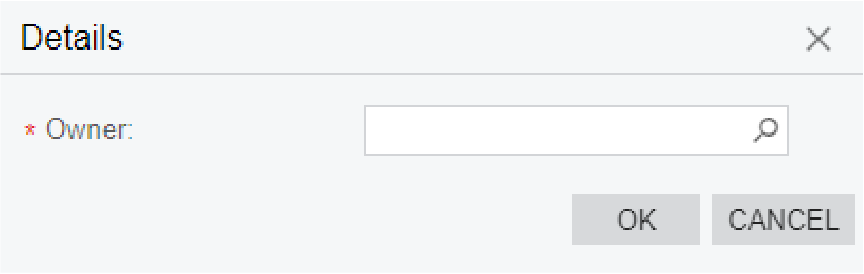


Figure: The Details dialog box for the Assign action



Lesson 3.4: Finalizing Customization of the Workflow

Learning Objectives

In this lesson, you will learn how to do the following:

- Use of the Workflow (Diagram View) to customize workflows.

Diagram View: To Remove Unneeded Parts of the Workflow

Acting as the technical specialist, you need to remove the transition and the action that are not required in the workflow of the Opportunities (CR304000) form anymore.

Diagram View: To Adjust the System State

Acting as the technical specialist, you need to make the Owner box required for the opportunities with the Open status on the Opportunities (CR304000) form.

Diagram View: To Test the Inherited Workflow

Acting as the technical specialist, you need to publish your customization project and then test your changes on the Opportunities (CR304000) form to make sure that the customized workflow works as expected.

Day 3

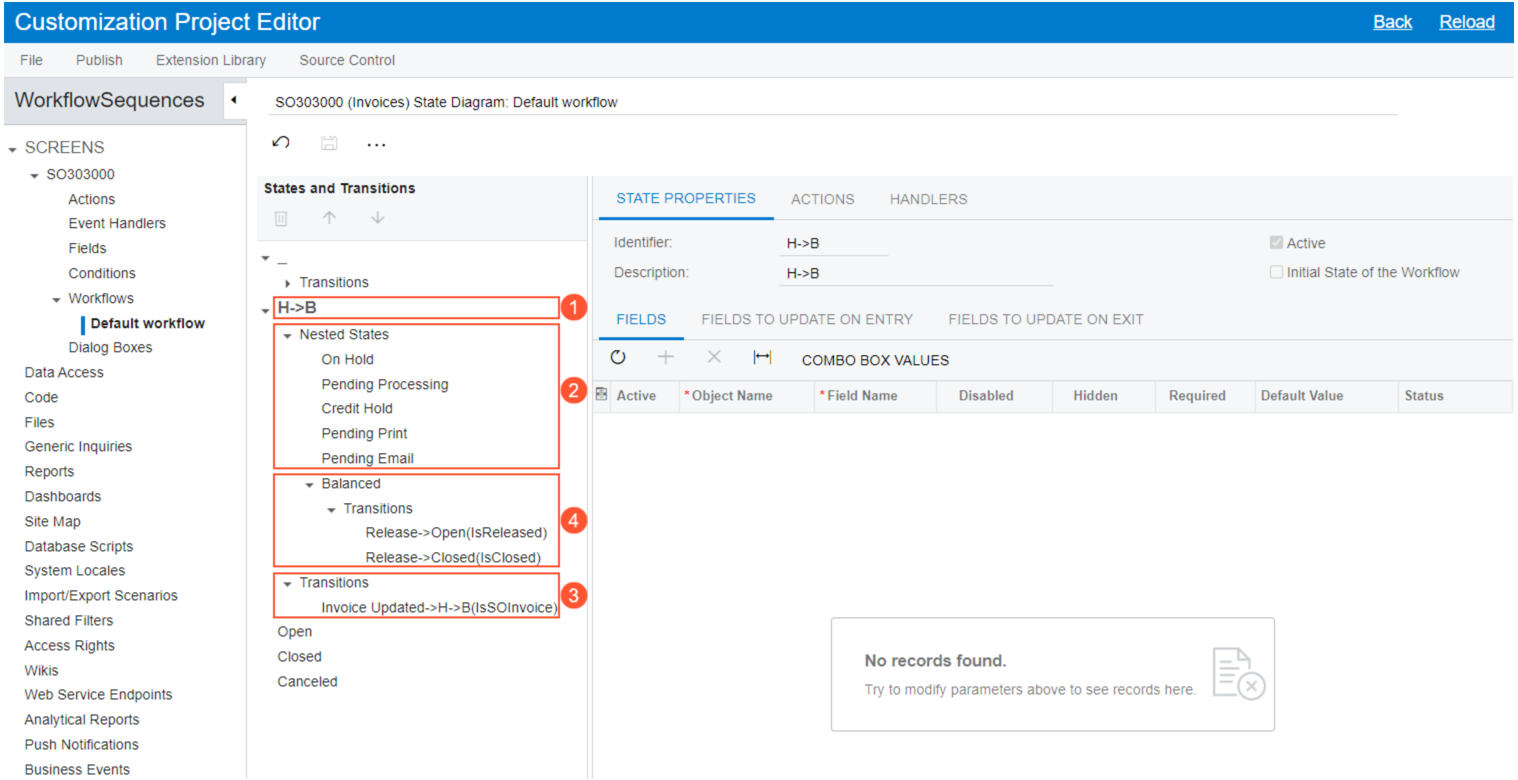
Lesson 3.5: Modifying a Workflow with a Composite State

Learning Objectives

In this lesson, you will learn how to do the following:

- What a composite state is and how to modify it in an inherited workflow.

Figure: A composite state in the invoice workflow



Composite States: To Modify a Workflow with a Composite State

On the Invoices (SO303000) form, the default workflow includes a composite state with multiple nested states. Suppose that you want to add a new state, Postponed, to the composite state in the workflow for the Invoices form after the Credit Hold state. Further suppose that you need to specify a skip condition for it and add a transition from this state to the next state in the workflow.

If a user specifies a discount for an invoice, you want this invoice to be reviewed and the discount to be approved before the user can proceed with processing the invoice. After the discount is approved, the system specifies the current date in the Cash Discount Date box. The skip condition will check for no cash discount being applied to the invoice. That is, if the user does not specify any discount, the system will skip the Postponed state.

Lesson 4.1: Creating an Inherited Workflow with a Workflow-Identifying Field

Learning Objectives

In this lesson, you will learn how to do the following:

- Creating a workflow that is based on the specific value of a selected field.

Figure: The Workflows page

SO301000 (Sales Orders) Workflows

ADD WORKFLOW ···

State Identifier: Status

Workflow-Identifying Field: Behavior

☐ Allow Users to Modify Value

Active	Workflow Type	Workflow Name	Base System Workflow	Status
☑	MO	MO workflow		System Readonly
☑	BL	BL workflow		System Readonly
☑	CM	CM workflow		System Readonly
☑	IN	IN workflow		System Readonly
☑	RM	RM workflow		System Readonly
☑	QT	QT workflow		System Readonly
☑	SO	SO workflow		System Readonly
☑	TR	TR workflow		System Readonly

Workflow-Identifying Fields: To Create an Inherited Workflow

Acting as the technical specialist, you need to create a customized workflow for sales orders on the Sales Orders (SO301000) form. This form uses the Behavior internal field as the workflow-identifying field. For sales orders, the Behavior field has the SO value, which is the type of the workflow for sales orders.

Lesson 4.2: Implementing Automated Transitions

Learning Objectives

In this lesson, you will learn how to do the following:

- Create workflow conditions with user-defined fields.

Workflow-Identifying Fields: To Add Conditions with User-Defined Fields

Acting as a technical specialist, you need to add the following conditions to the workflow of sales orders on the Sales Orders (SO301000) form:

- TotalMoreThan800, which will be used to automatically put a sales order on hold if its OrderTotal is greater than $800 and the sales order has not been yet reviewed
- TotalLessThan800, which will be used to automatically remove a sales order from hold if its OrderTotal is less than $800 and it has not been put on hold manually

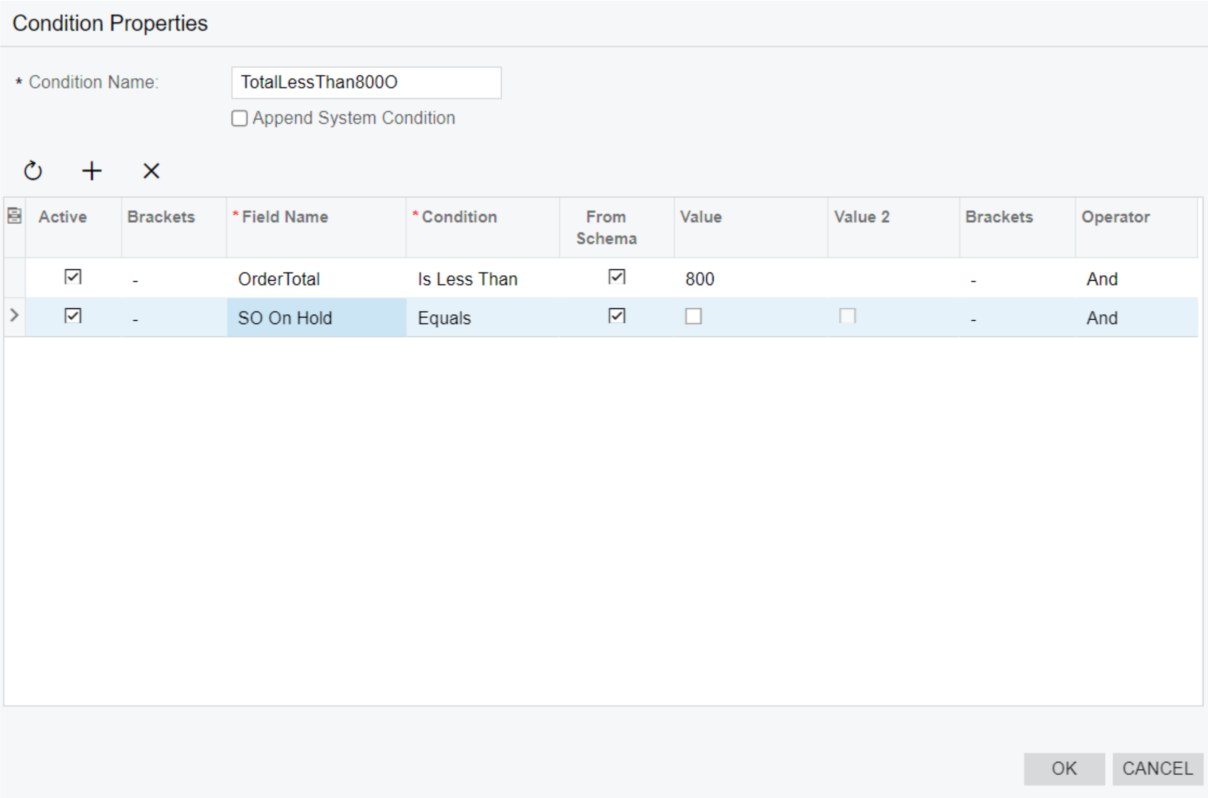
Because the system does not store information about whether the sales order has been reviewed or whether it has been put on hold manually, you need to create user-defined fields for the workflow of sales orders with the SO type. One field will be used to check if a sales order has been put on hold manually, and another will be used to check whether it has already been removed from hold. Because these fields will be used only internally, you will define them to be hidden.

Figure: The SO30100 (Sales Orders) Fields page

SO301000 (Sales Orders) Fields

Object Name	Field Name	Disabled	Hidden	Required	Display Name	From Schema	Default Value	Status
PX.Objects.SO.SOOrder	AttributeSOONHOLD		True		SO On Hold	☐		New
PX.Objects.SO.SOOrder	AttributeSOREVIEW		True		SO Reviewed	☐		New
PX.Objects.SO.SOOrder	Status				Status	☐		Inherited

Figure: The TotalLessThan800 condition



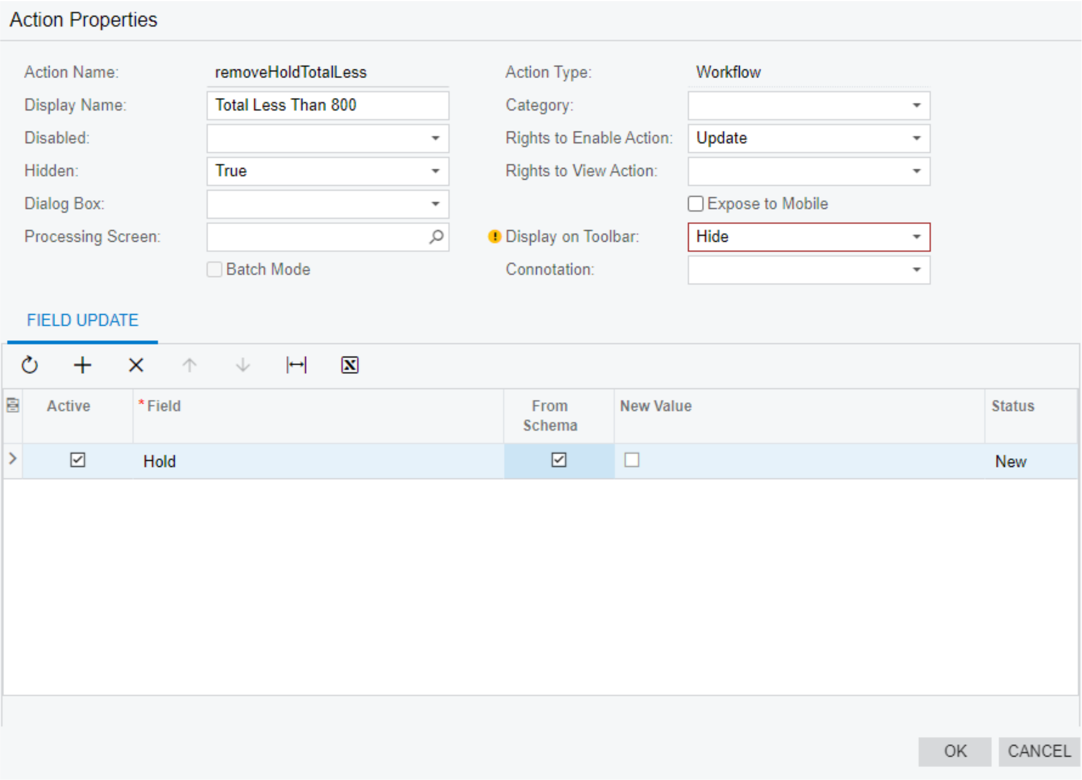
Workflow-Identifying Fields: To Automate Transitions by Using Conditions (in the Tree View)

In the predefined sales order workflow, when a user clicks Remove Hold on the form toolbar or More menu of the Sales Orders (SO301000) form, the releaseFromHold action triggers the transition from the On Hold state to one of the following states: Open, Pending Processing, or Awaiting Payment.

In the customized workflow, acting as a technical specialist, you need to add new transitions from the On Hold state to each of these states. The added transitions should be performed if the OrderTotal is less than $800—that is, a sales order should be automatically removed from hold only if this condition is met. With these changes implemented, a sales order on the Sales Orders form will remain on hold only if its OrderTotal is greater than $800 or if a user puts it on hold manually. In all other cases, the sales order will be removed from hold automatically.

If a sales order is removed from hold automatically, it should be moved to the same state (depending on which conditions are fulfilled) to which it would have moved if it had been removed from hold manually.

Figure: The Action Properties dialog box



Workflow-Identifying Fields: To Automate Transitions by Using Conditions (in the Diagram View)

In the predefined sales order workflow, when a user clicks Hold on the form toolbar or More menu of the Sales Orders (SO301000) form, the Hold (putOnHold) action triggers transitions to the On Hold state from any of the following states:

- Credit Hold
- Pending Processing
- Awaiting Payment
- Open
- Back Order

Suppose that you are a technical specialist. In the customized workflow, you need to add new transitions to the On Hold state from each of these states. With the changes implemented, a sales order will be put on hold automatically from any of these states if its OrderTotal on the Sales Orders (SO301000) form is greater than or equal to $800 and it has not been removed from hold manually.

Lesson 4.3: Modifying System Actions and Testing the Customization Project

Learning Objectives

In this lesson, you will learn how to do the following:

- Make system actions modify custom fields.

Workflow-Identifying Fields: To Make System Actions Modify Custom Fields

In the inherited workflow, if a sales order has been manually removed from hold once, it should not be possible to put this sales order on hold again, even if its OrderTotal has been increased. Therefore, when a user clicks the Hold or Remove Hold button on the Sales Orders (SO301000) form, you need to indicate in the system that the sales order has been manually put on hold or removed from hold, respectively. To save this information, you will use the custom fields that you have added in Workflow-Identifying Fields: To Add Conditions with User-Defined Fields. The system actions should update the values of these fields.

Workflow-Identifying Fields: To Test the Inherited Workflow with a Workflow-Identifying Field

Acting as the technical specialist, you need to publish your customization project and then test your changes on the Sales Orders (SO301000) form to make sure that the inherited workflow works as expected.

Lesson 5.1: Upgrading Workflows

Learning Objectives

In this lesson, you will learn how to do the following:

- To include changes in the predefined workflow in a customization project that contains an inherited workflow that is based on this predefined workflow.

Upgrade of Workflows: To Upgrade a Customization Project That Contains an Inherited Workflow

Suppose that after you have created an inherited workflow for the Opportunities (CR304000) form, Acumatica developers have modified the predefined workflow for this form: In the predefined workflow, the Review state and the Need Review action have been added. (In this activity, you will load the changes to the predefined workflow by importing and publishing the OpportunitySystemWorkflowChanges.zip customization package.)

Acting as a technical specialist, you need to upgrade the version of the predefined workflow in the customization project so that both the predefined workflow that is included in the customization project and the inherited workflow that is based on this predefined workflow contain the changes introduced by Acumatica developers.

Figure: The message about conflicts during upgrade

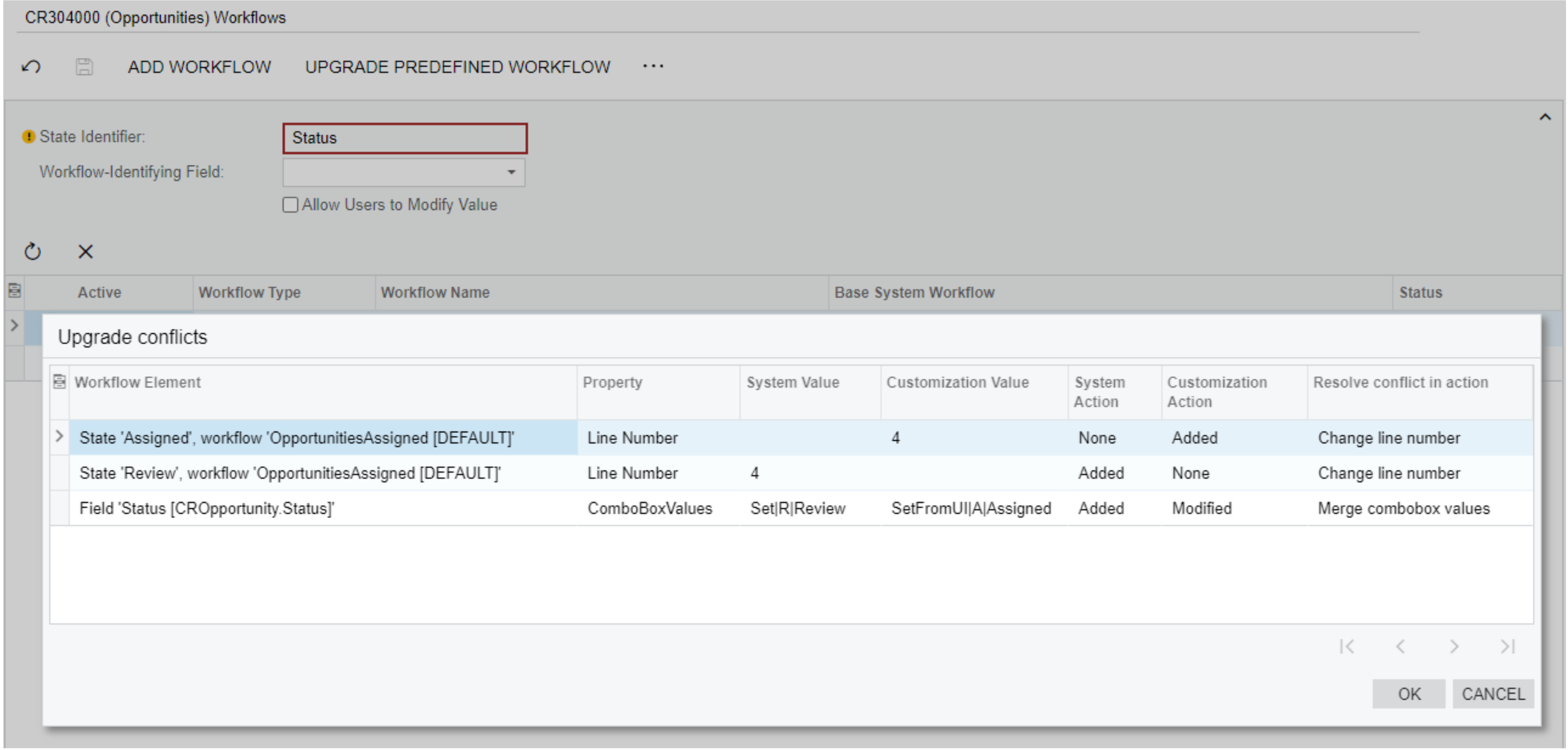
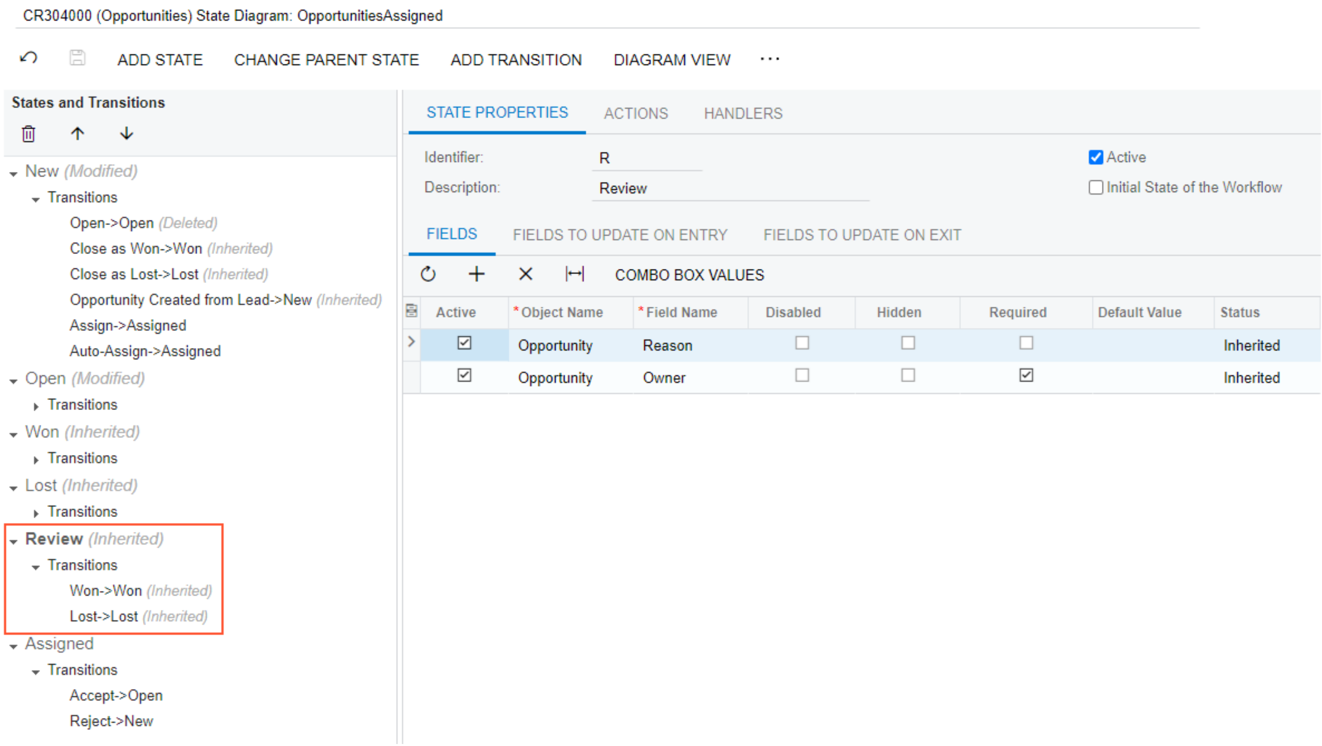


Figure: The upgraded inherited workflow



Thank you!

Nicholas Bova