Acumatica
The Cloud ERP

T270 Workflow APIs

Varthini Bhaskaran

Technical Account Manager

Timing and Agenda

October 31, 2022 -10 -11 AM

Day 1

Lesson 1.1: Set Up the Basic Parts of the Workflow

November 1, 2022 -10 -11 AM

Day 2

Lesson 1.2: Implement a Simple Transition

Lesson 1.3: Implement a Group of Transitions

Lesson 1.4: Implement a Transition with a Dialog Box

Timing and Agenda

November 2, 2022 -10 -11 AM

Day 3

Lesson 1.5: Implement a Transition with Field Assignments

Lesson 2.1: Use an Existing Event in a Custom Workflow

November 3, 2022 -10 -11 AM

Day 4

Lesson 2.2: Create a New Event

Lesson 3.1: Customize the Workflow for Invoices

Day 1

Part 1: Creating a Custom Workflow

Joe Gibbs Racing
Acumatica Partner

Lesson 1.1: Set Up the Basic Parts of the Workflow

Learning Objectives

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

- Determine the set of states of a document in the workflow
- Define the set of states for the workflow
- Create the screen configuration method
- Enable workflow validations

Lesson Summary

In this lesson, you have prepared the base components of the workflow. You have learned how to create a workflow class, add a screen configuration, and define the set of states for the workflow.

Day 2

Lesson 1.2: Implement a Simple Transition

Learning Objectives

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

- Define and configure workflow actions
- Define a default category for workflow actions
- Define and configure workflow states
- Define a simple transition

Figure: A new document on the Repair Work Orders form

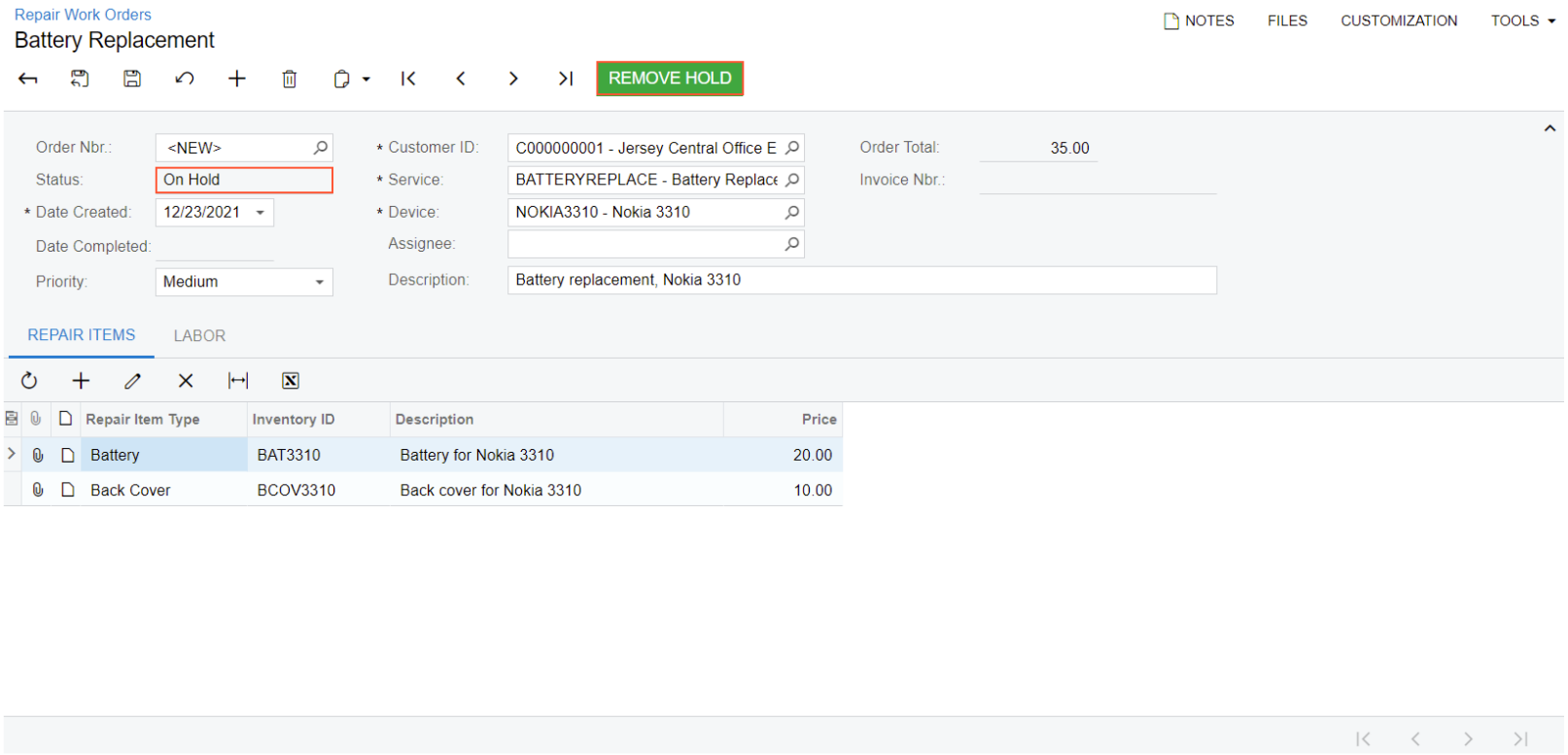
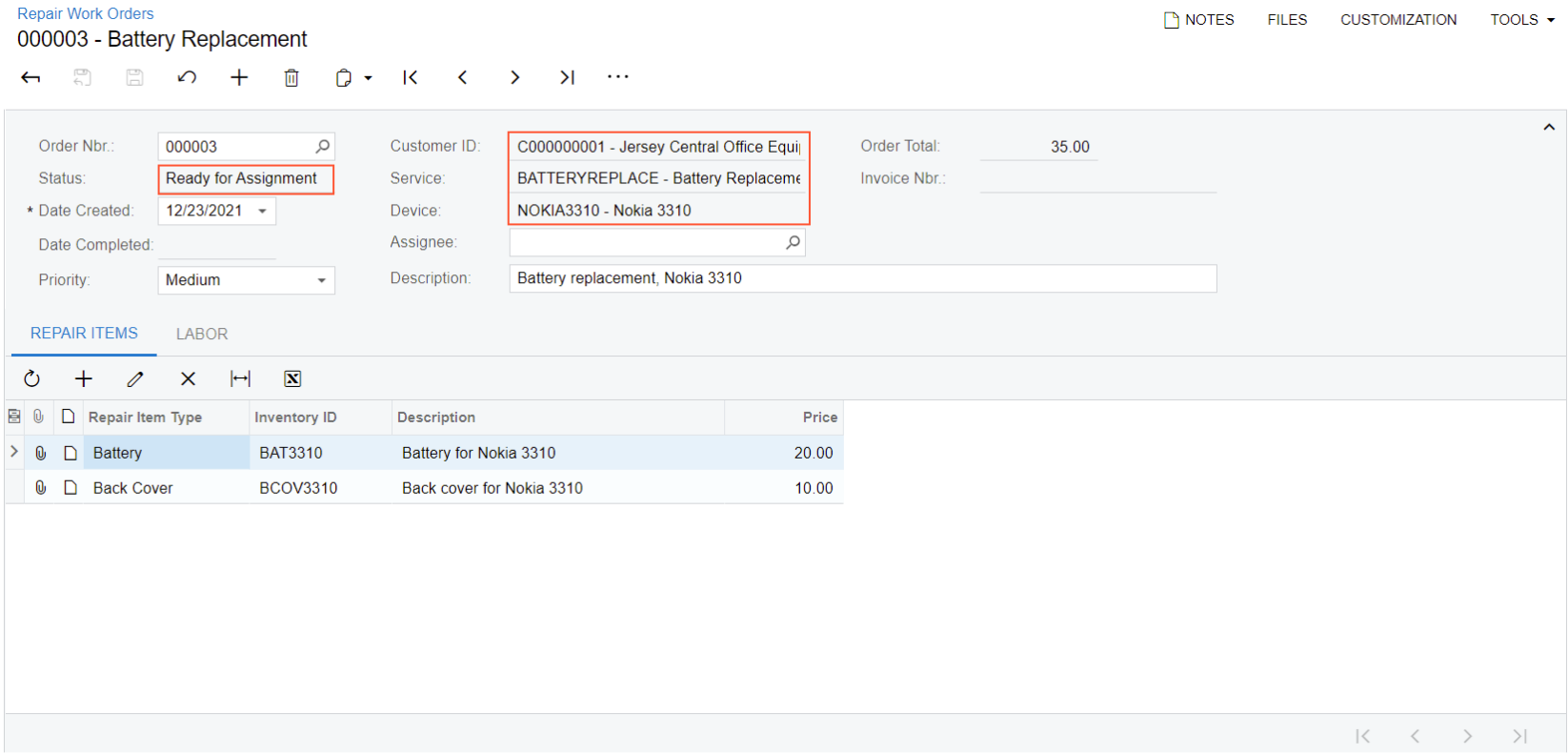


Figure: A document with the Ready for Assignment status



Lesson Summary

In this lesson, you have implemented the first transition for the workflow of the Repair Work Orders (RS301000) form. You have learned how to define workflow actions, workflow states, and a transition. You have also tested the transition.

Lesson 1.3: Implement a Group of Transitions

Learning Objectives

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

- Define a condition in a workflow
- Specify a condition for a transition
- Unite transitions into a group

Lesson Summary

In this lesson, you have implemented a transition from the OnHold state to the PendingPayment state and added conditions for previously implemented transitions. You have learned how to put transitions into groups and declare condition packs.

Exercise 1.1: Implement the PutOnHold Transition

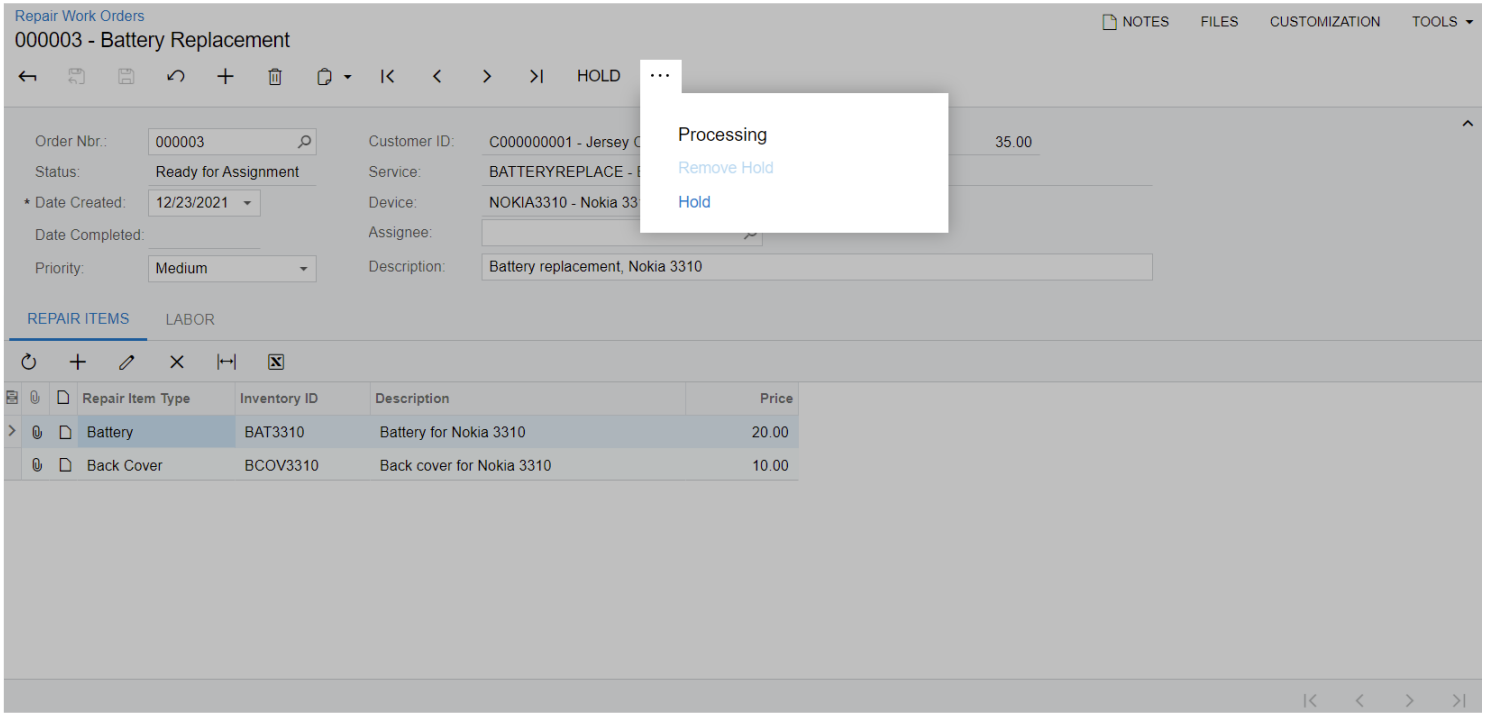


Figure: The More menu of the Repair Work Orders form

Lesson 1.4: Implement a Transition with a Dialog Box

Learning Objectives

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

- Define a dialog box in a workflow
- Specify a dialog box for an action

Figure: The Assign button

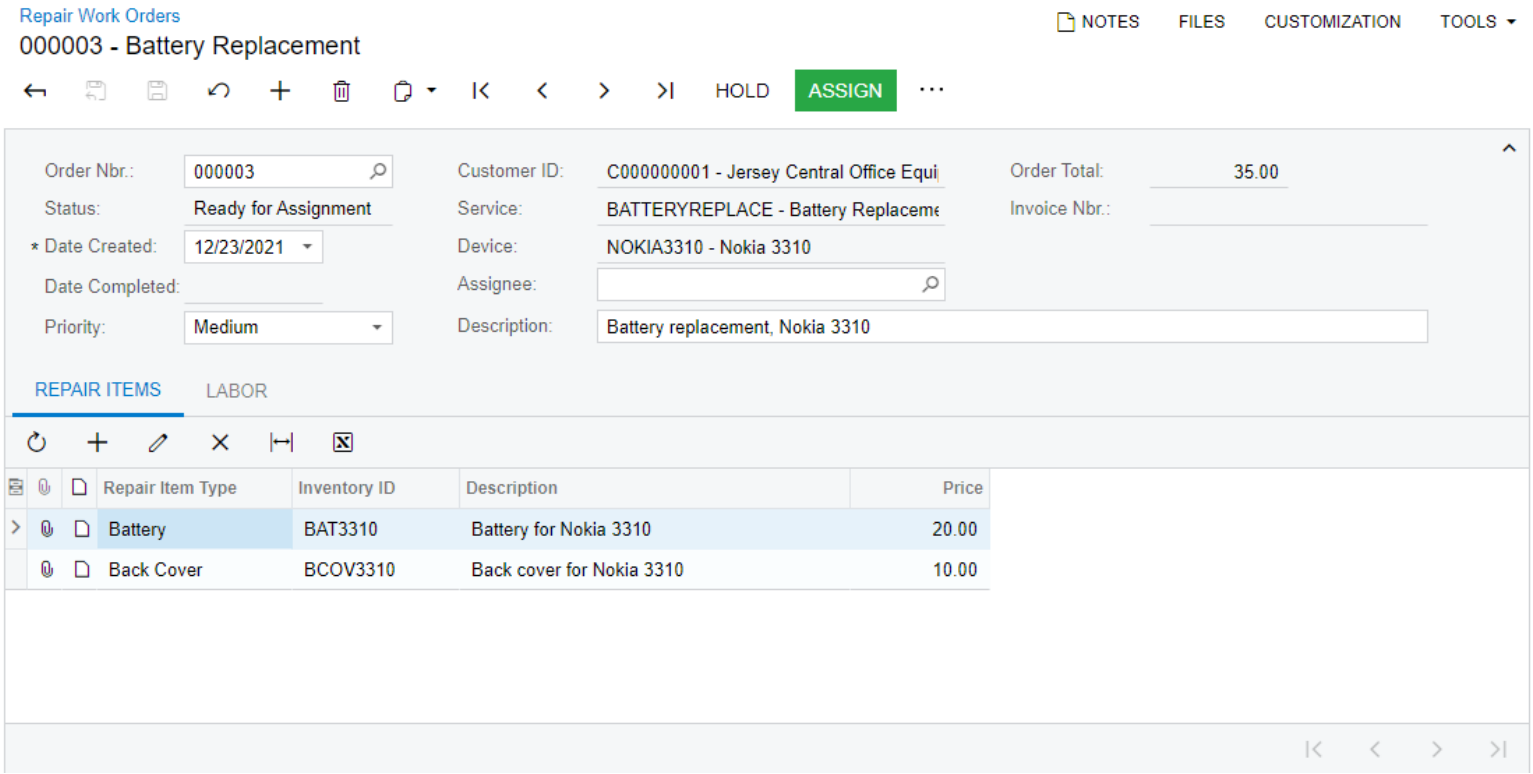
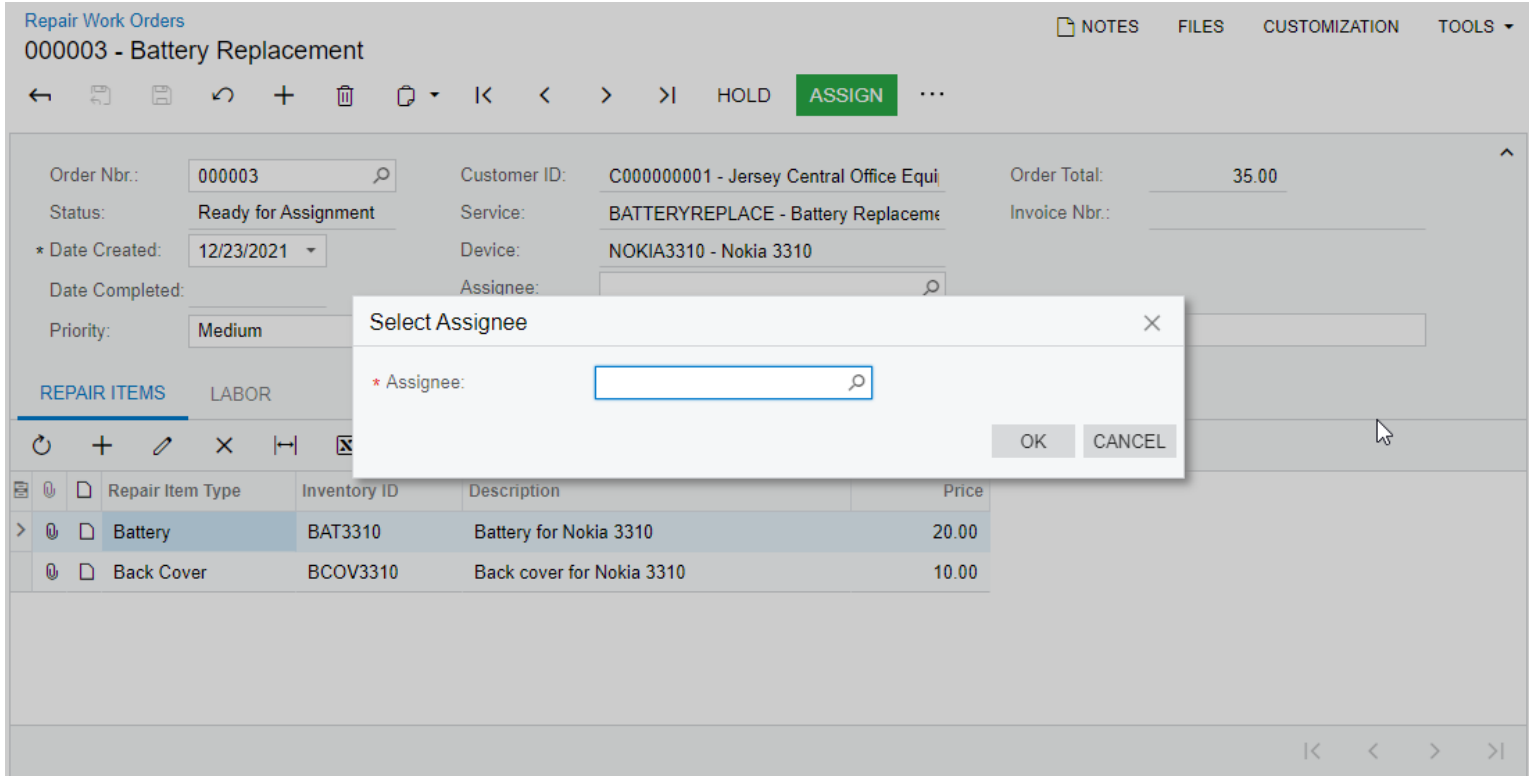


Figure: The Select Assignee dialog box



Lesson Summary

In this lesson, you have implemented the Assign action, a transition from the ReadyForAssignment state to the Assigned state, and a dialog box that is shown when a user clicks the Assign button on the form toolbar.

Day 3

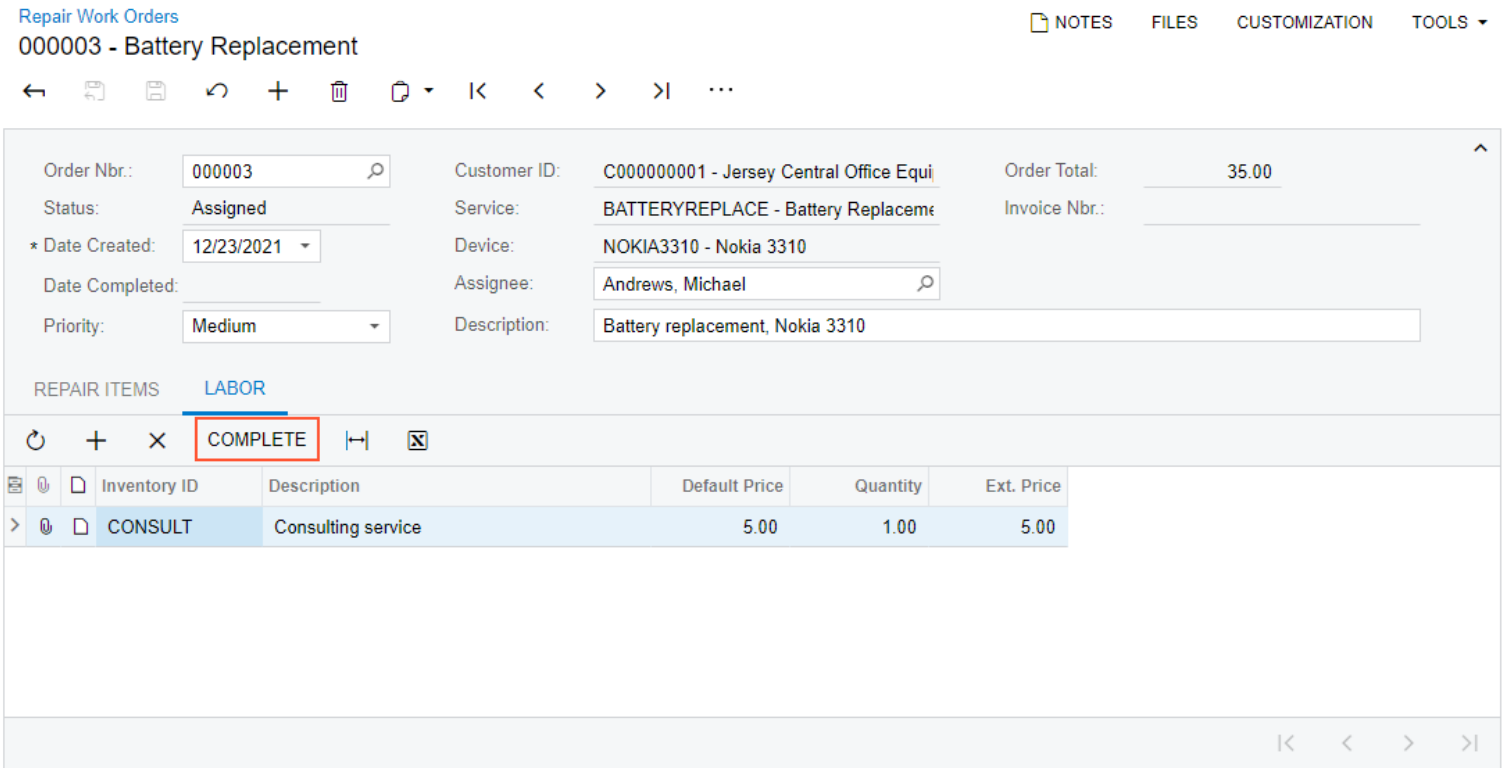
Lesson 1.5: Implement a Transition with Field Assignments

Learning Objectives

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

- Define a DAC field assignment with the workflow API methods
- Define the location of the button and command associated with a workflow action

Figure: The Complete button



Lesson Summary

In this lesson, you have implemented the Complete action, as well as its associated button on the table toolbar and command on the More menu. You have also defined the transition from the Assigned state to the Completed state.

Figure: Creation of an invoice

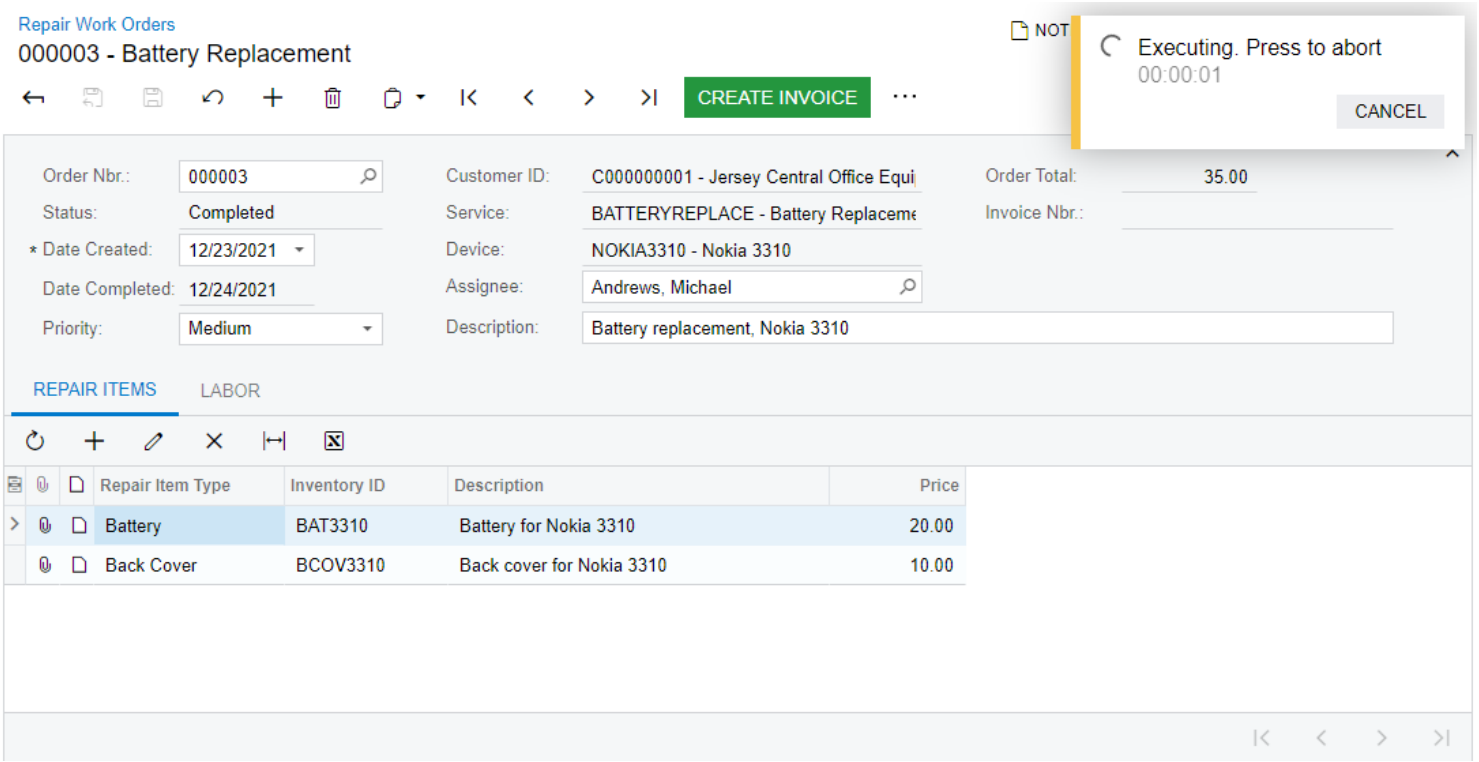
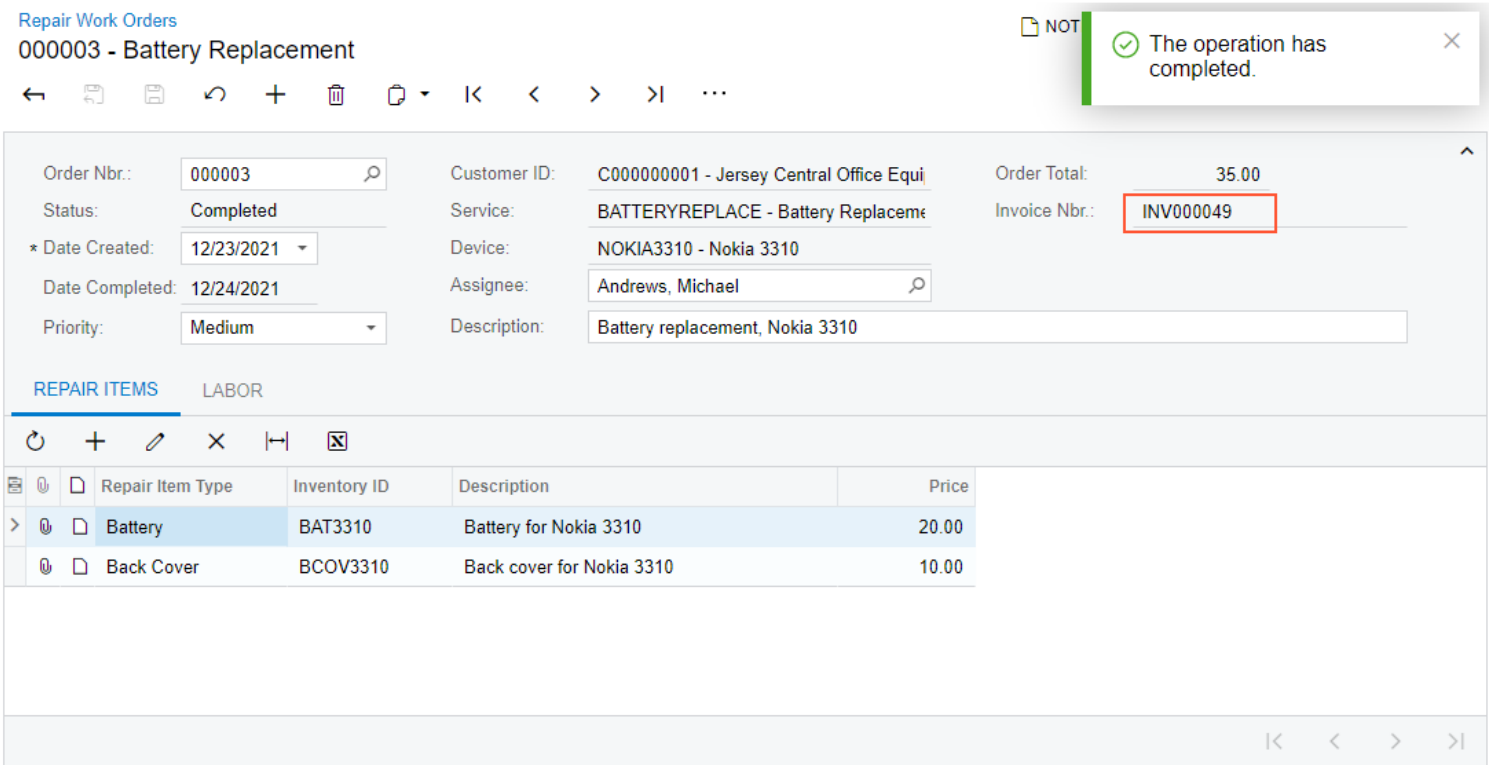


Figure: The invoice number of the repair work order



Part 1 Summary

In the part of the course, you have learned how to define a screen configuration. You have learned how to define workflow states, actions, dialog boxes, and transitions. You have also learned how to configure these parts of the screen configuration.

Part 2: Incorporating an Existing Workflow into a Custom Workflow

Joe Gibbs Racing
Acumatica Partner

Lesson 2.1: Use an Existing Event in a Custom Workflow

Learning Objectives

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

- Explore the code of the predefined workflow to find the event
- Create a custom event handler
- Bind the event handler to an event
- Register an event handler in a particular state
- Implement a transition triggered by an event
- Override the Persist method

Figure: The cleared the Enable Just My Code check box

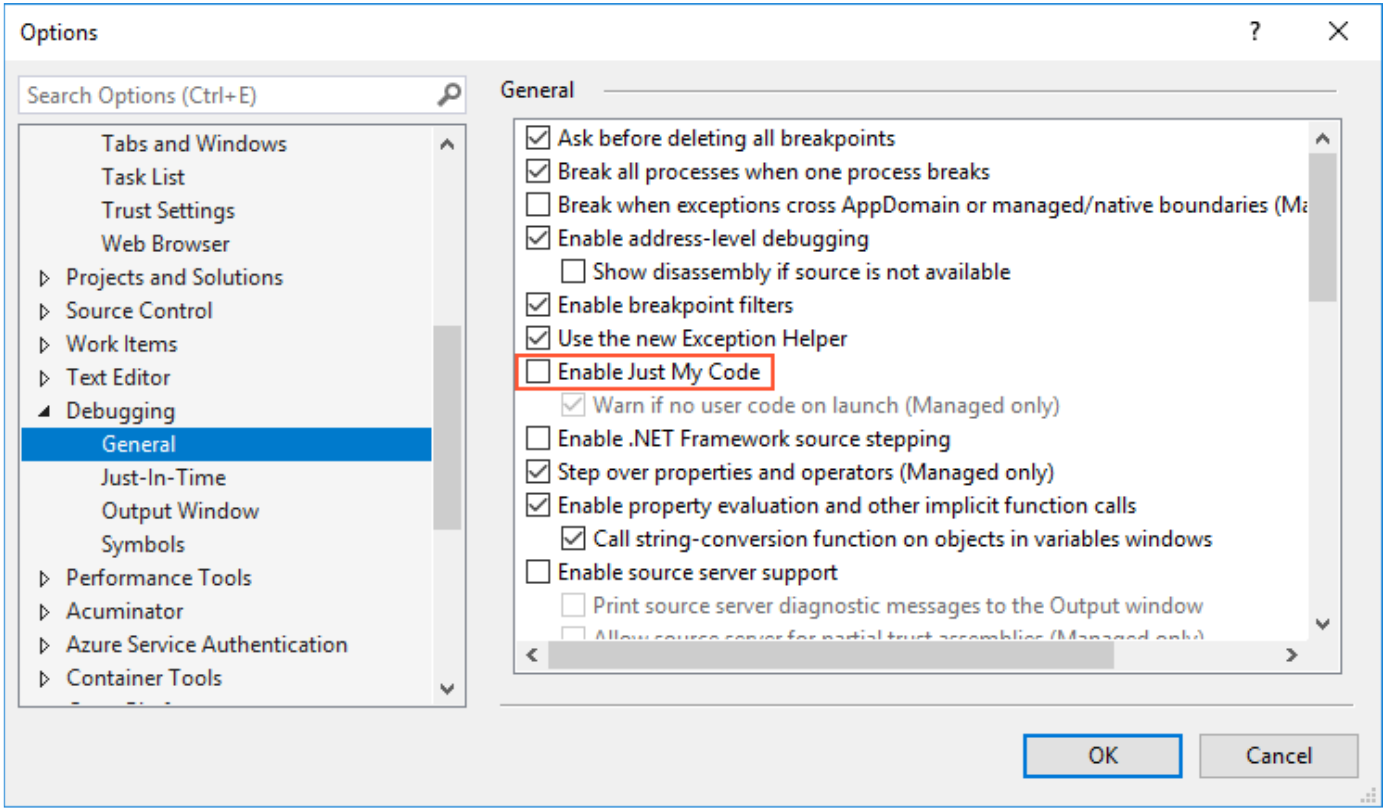


Figure: The source code of the Release action

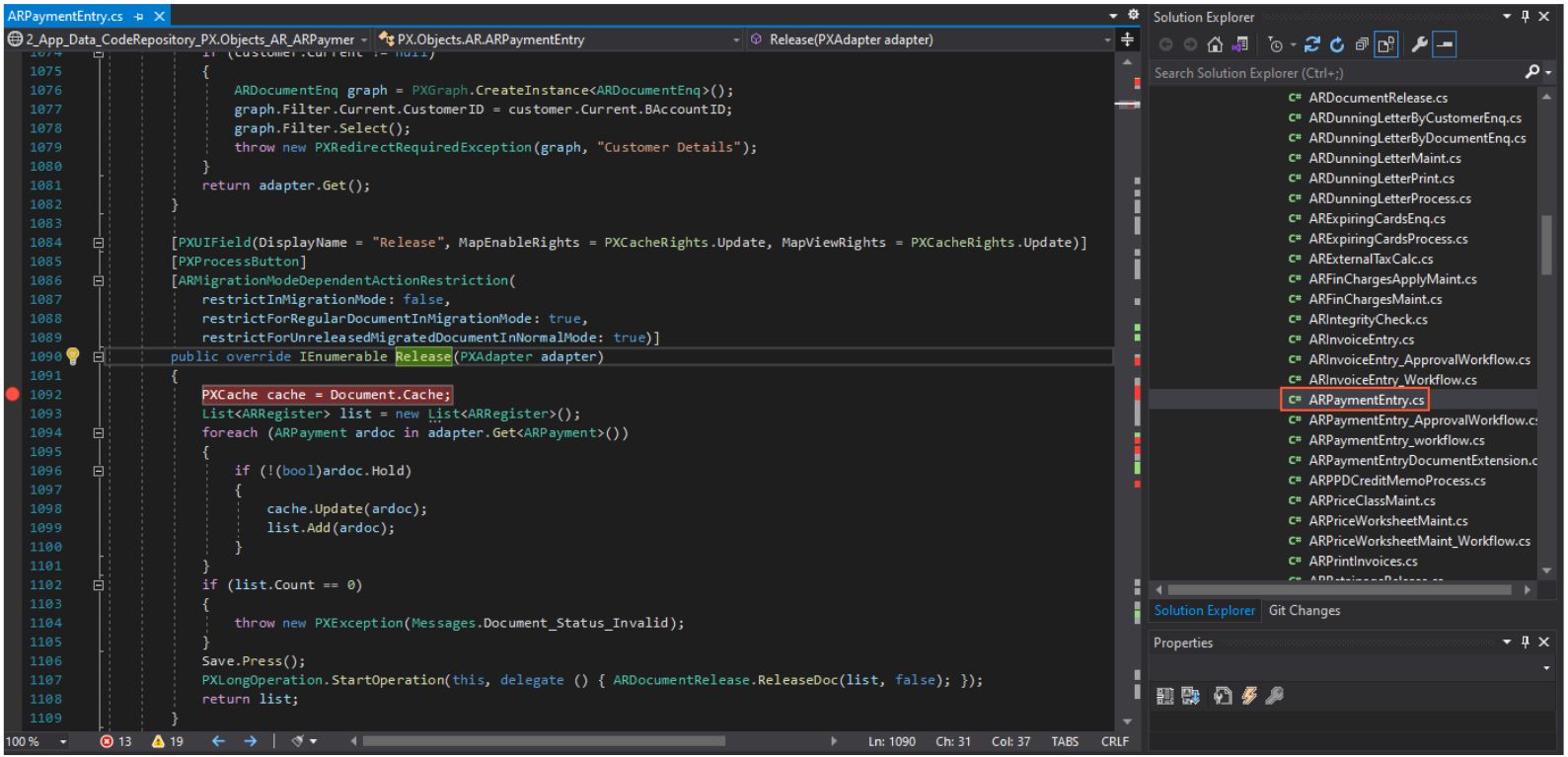


Figure: The call hierarchy for the CloseInvoiceAndClearBalances method

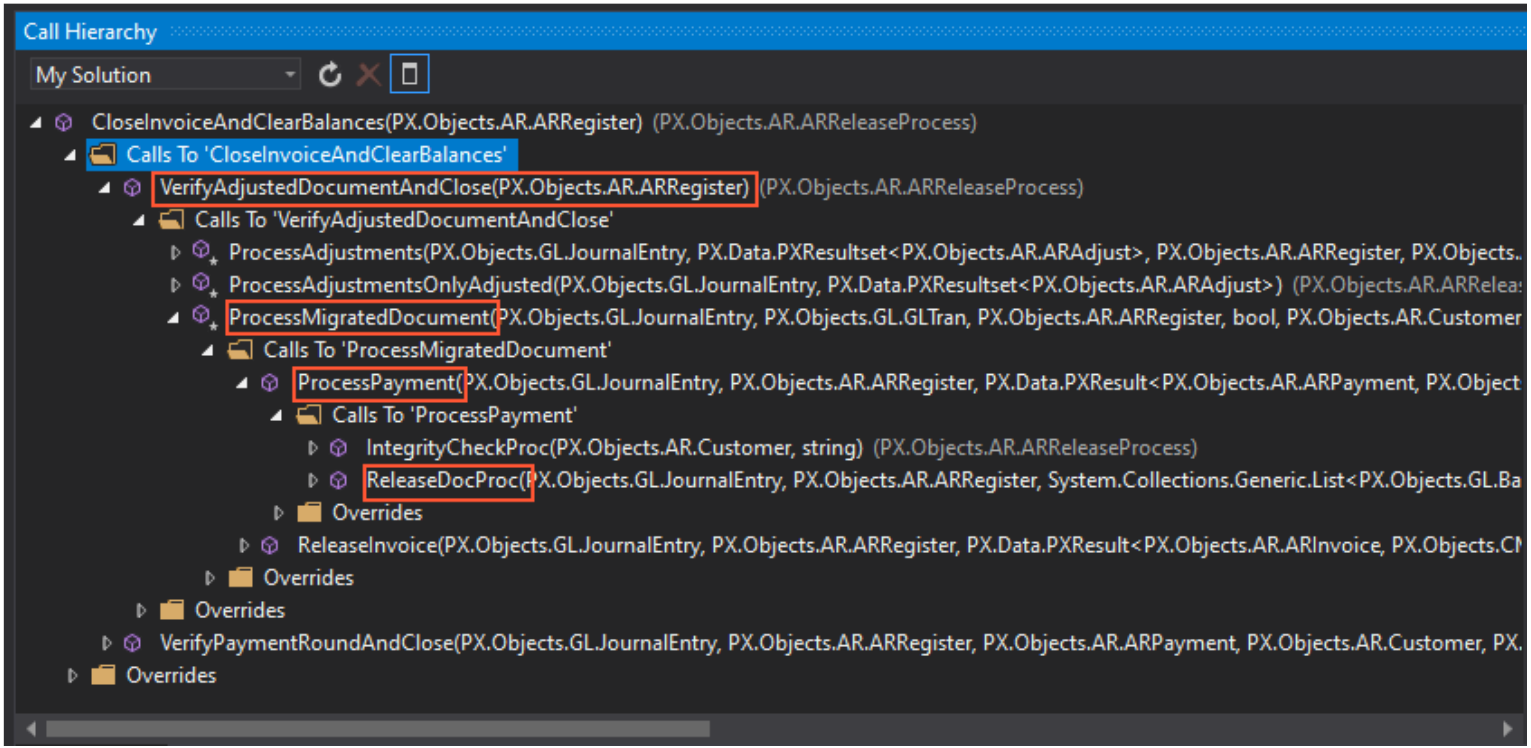
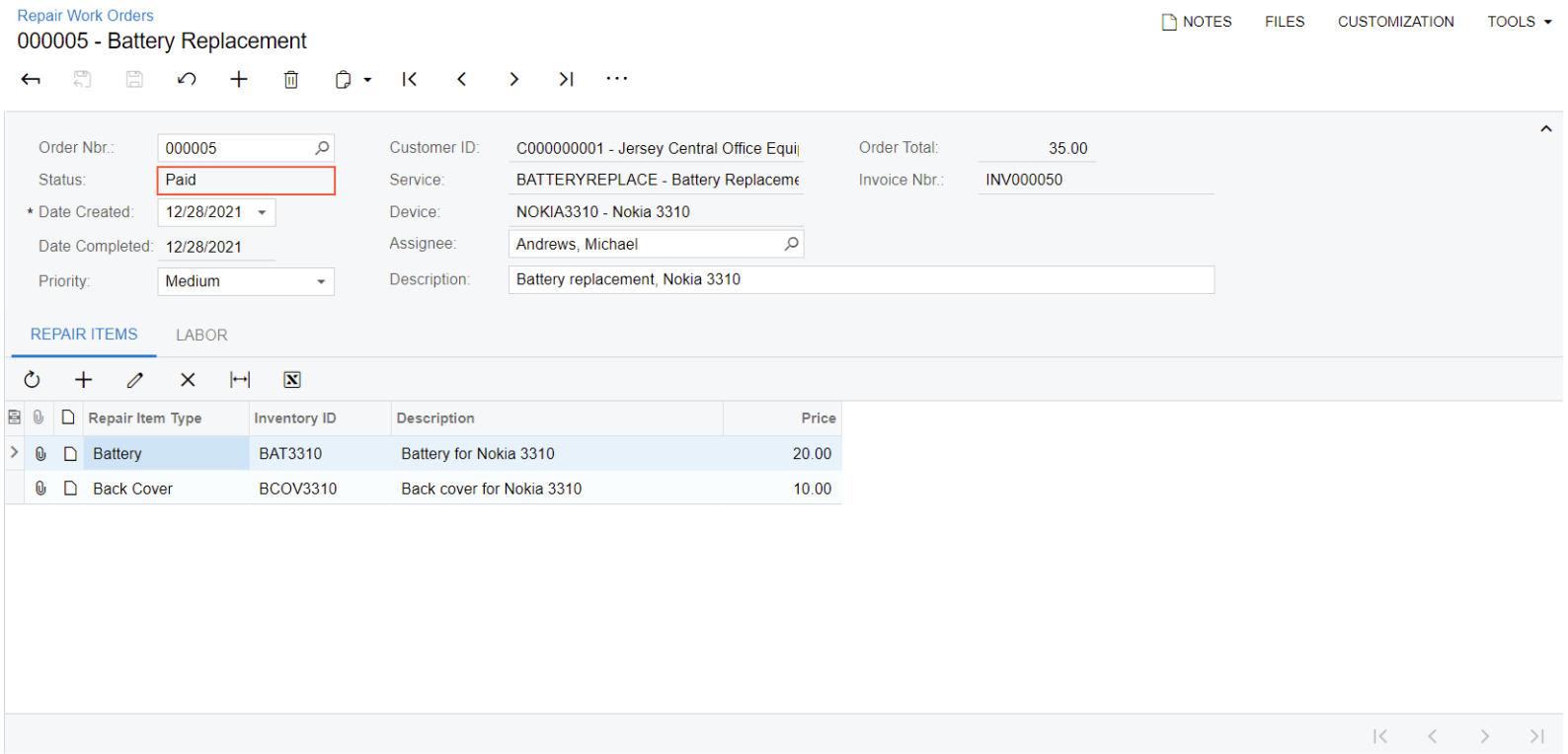


Figure: Transitions from the Open state

```
transitions.AddGroupFrom<State.open>(ts =>
{
    ts.Add(t => t
        .To<State.open>()
        .IsTriggeredOn(g=>g.emailInvoice)
        .WithFieldAssignments(fass => fass.Add<emailed>(v => v.SetFromValue(true))));
    ts.Add(t => t
        .To<State.closed>()
        .IsTriggeredOn(g => g.OnReleaseDocument)
        .When(conditions.IsClosed));
    ts.Add(t => t
        .To<State.closed>()
        .IsTriggeredOn(g => g.OnCloseDocument));
    ts.Add(t => t
        .To<State.canceled>()
        .IsTriggeredOn(g => g.OnCancelDocument));
});
```

Figure: The repair work order with the Paid status



Lesson Summary

In this lesson, you have used the OnCloseDocument event to trigger the transition from the Completed state to the Paid state. You have implemented the Paid state, created a custom event handler, and registered the event handler in the screen configuration. You have also overridden the Persist method to save changes to the database.

Day 4

Lesson 2.2: Create a New Event

Learning Objectives

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

- Derive the value for a custom field from another form
- Create a custom event
- Override a method to fire the event

Figure: The prepayment for the created invoice

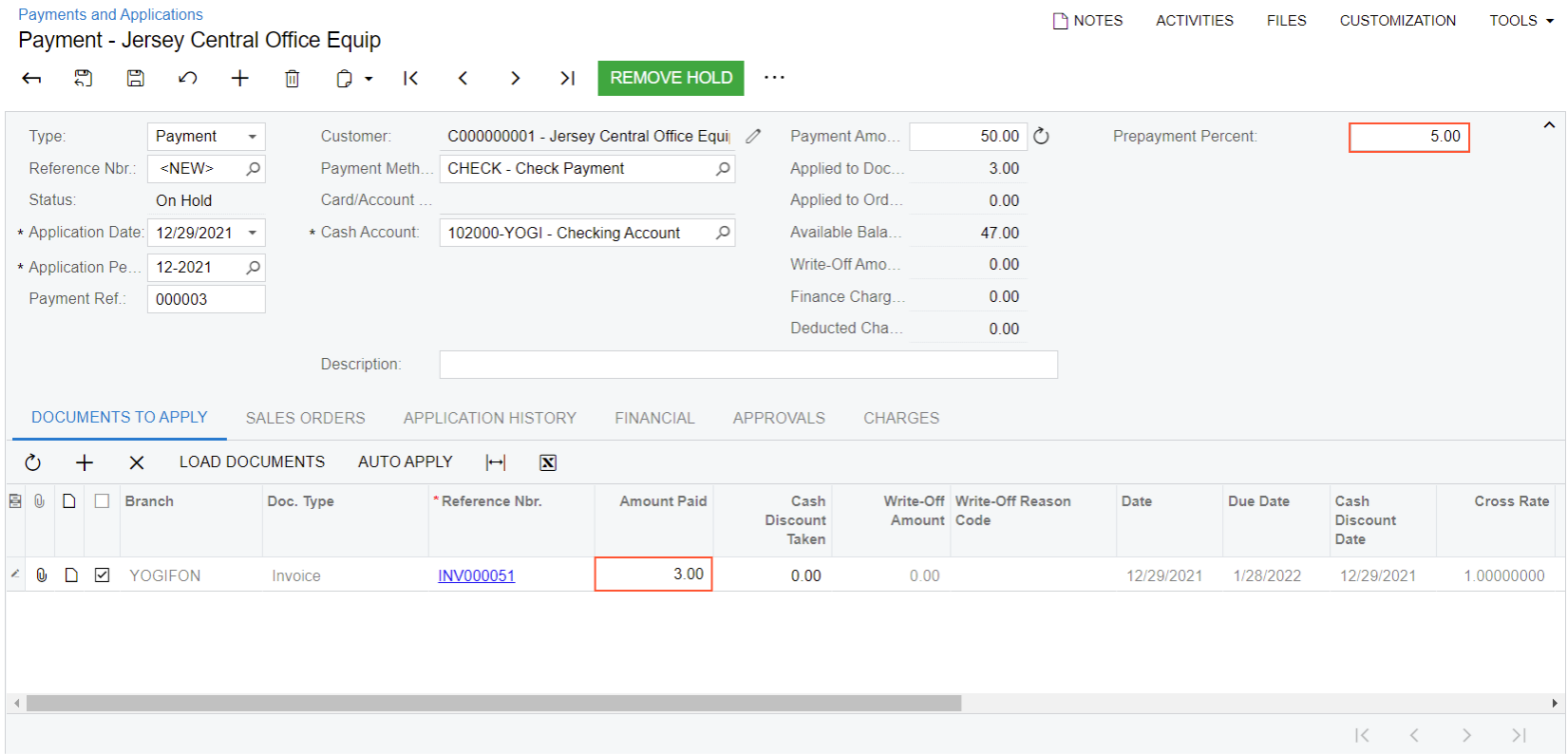
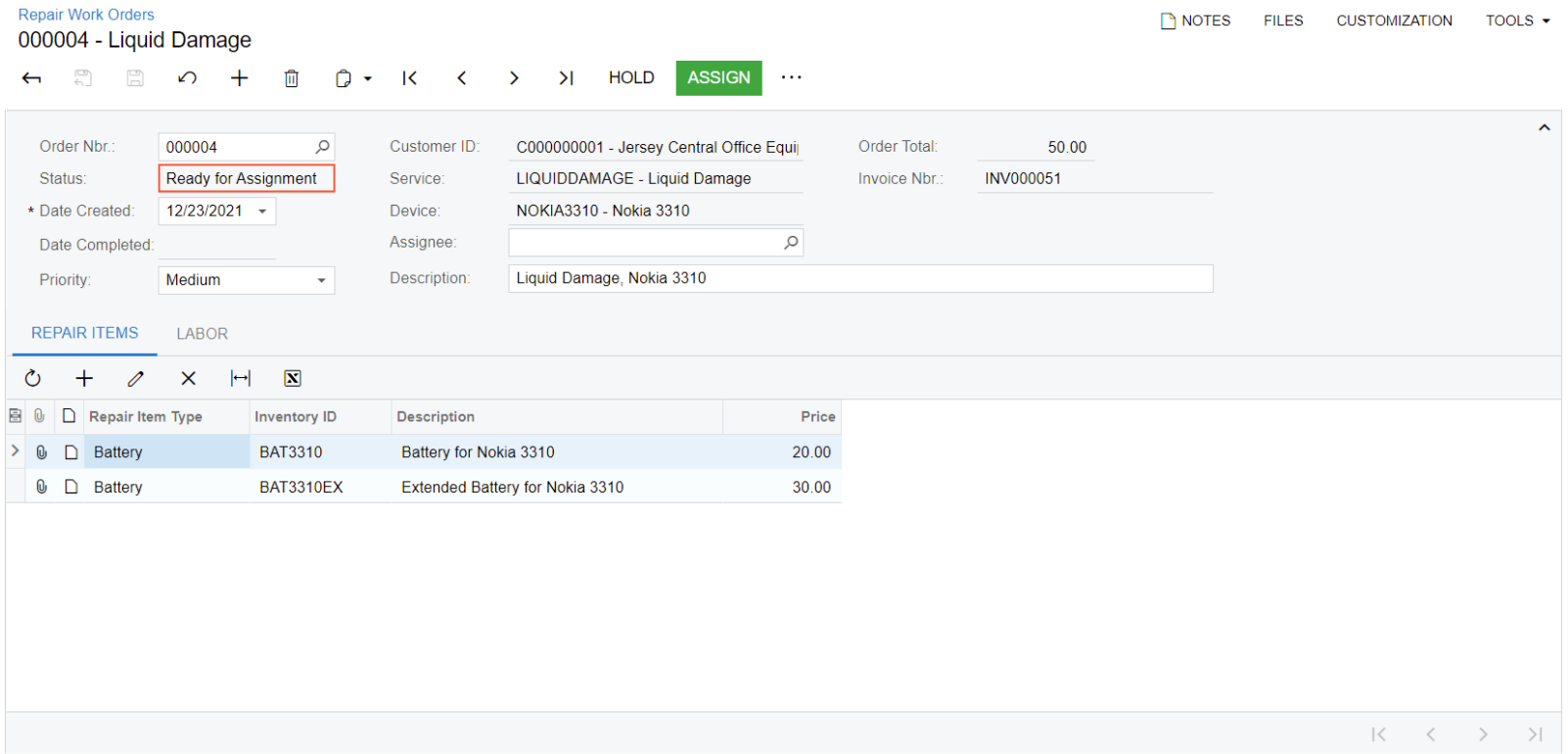


Figure: The repair work order with the Ready for Assignment status



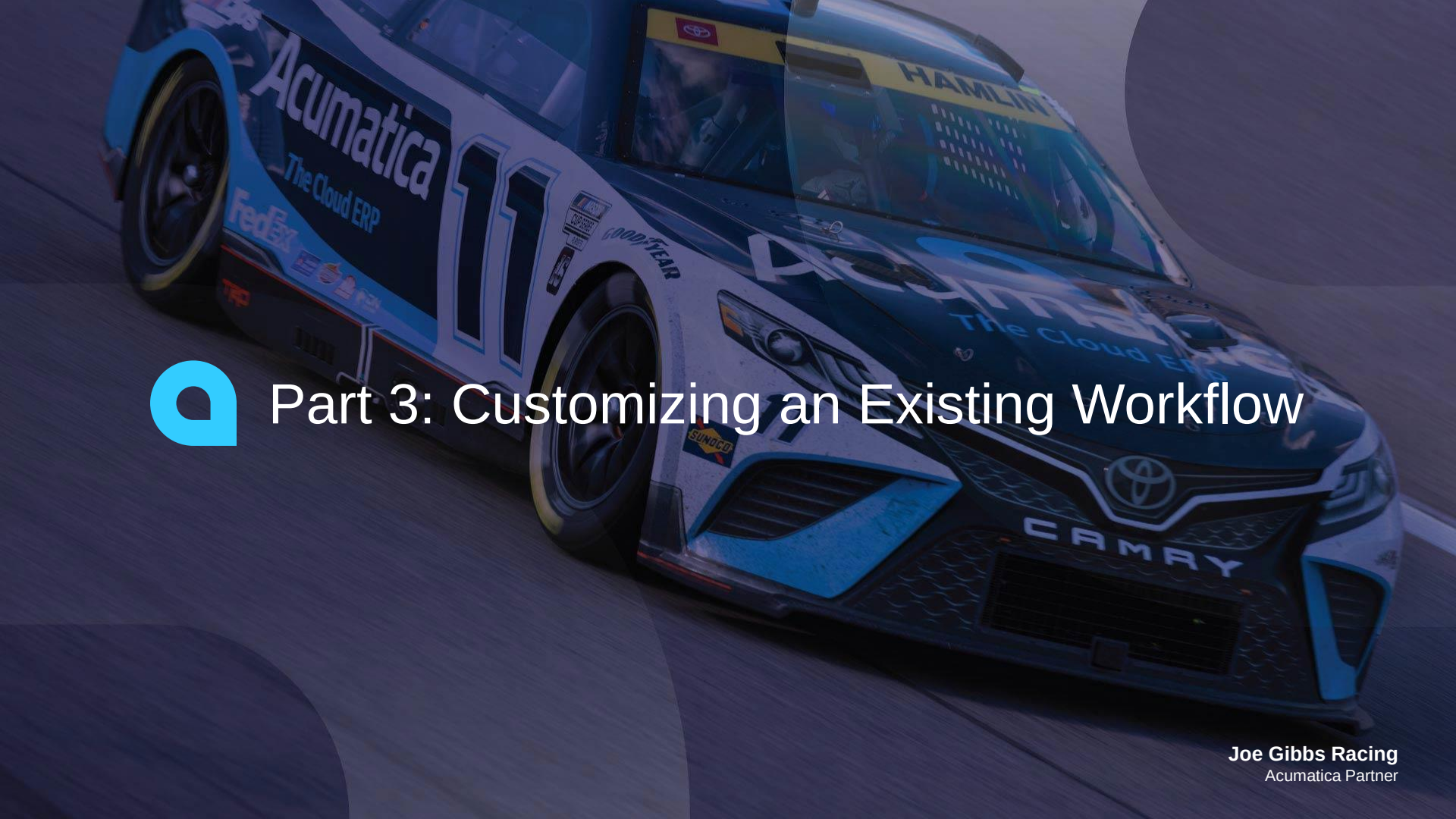
Lesson Summary

In this lesson, you have learned how to create a custom workflow event. You have implemented a new event, defined an event handler for it, and fired the event in an overridden method. You have also learned how to derive values from one entity and copy them to another entity.

Part 2 Summary

In this part, you have learned how to define a custom workflow event and fire it and how to use existing workflow events in a custom workflow. You have also learned how to derive a value of a custom field from another entity and how to explore the source code of Acumatica ERP.

You have completed the implementation of the workflow for the Repair Work Orders (RS301000) form. To verify that all elements of the workflow have been implemented, you can open the state diagram for the Repair Work Orders form and compare it to the one shown in Customization Description.



Part 3: Customizing an Existing Workflow

Joe Gibbs Racing
Acumatica Partner

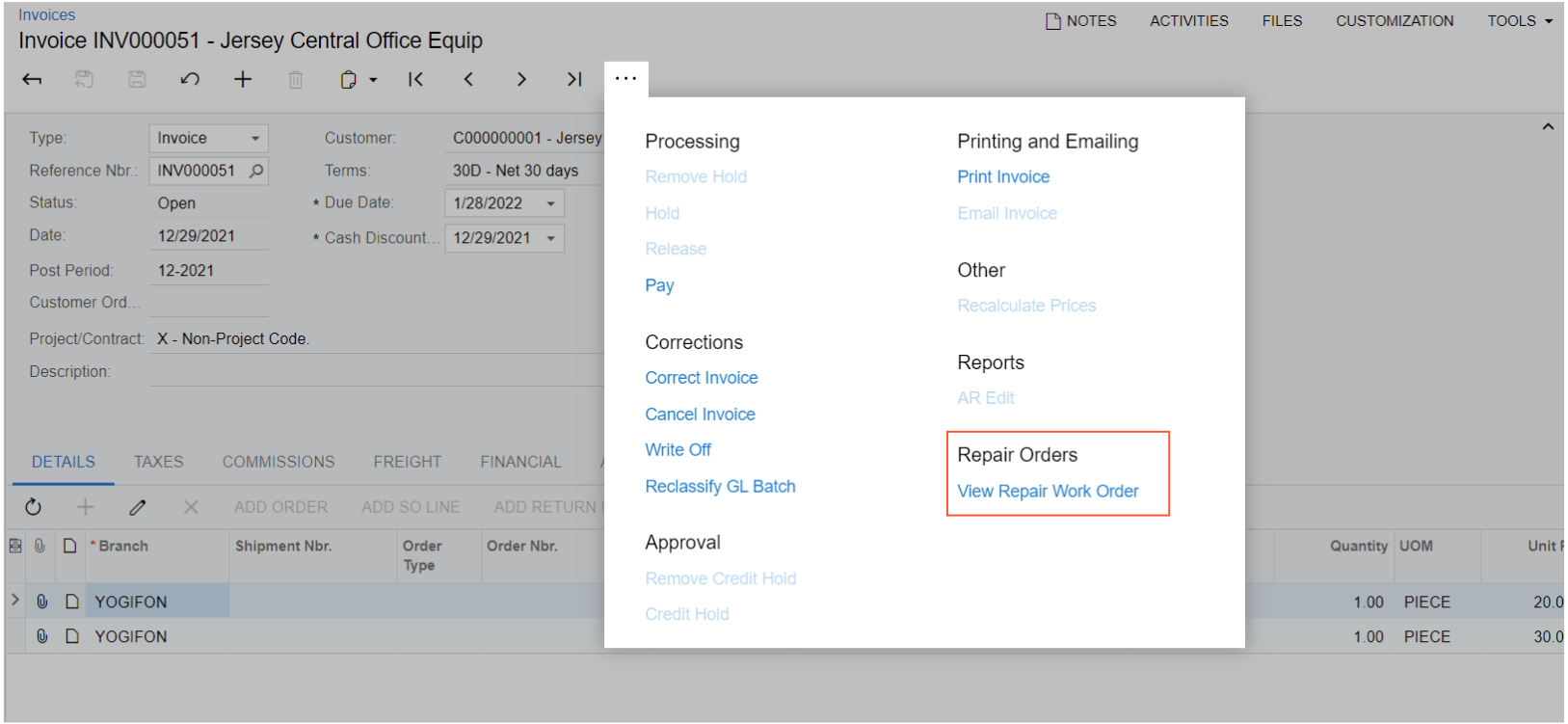
Lesson 3.1: Customize the Workflow for Invoices

Learning Objectives

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

- Update an existing screen configuration
- Define a new action category
- Define a workflow action based on a graph action

Figure: The More menu of the Invoices form



Lesson Summary

In this lesson, you have learned how to customize an existing workflow. You have implemented a new action and a new action category, and added them to the customized workflow.

Acumatica
The Cloud ERP

Thank you

Varthini Bhaskaran