



Workflow Engine

Customize Business Logic using UI or Code

Doug Johnson, Alexander Nesvizhsky

Agenda

- Workflow Engine History and Overview
- Demonstrations:

UI Demo: Adding inspection to purchase receipts

• Developer Demo: Financial Scenario

- Customer Examples
- What's New and What's Coming





Workflow Engine History and Overview



Workflows in Acumatica

Workflow Options

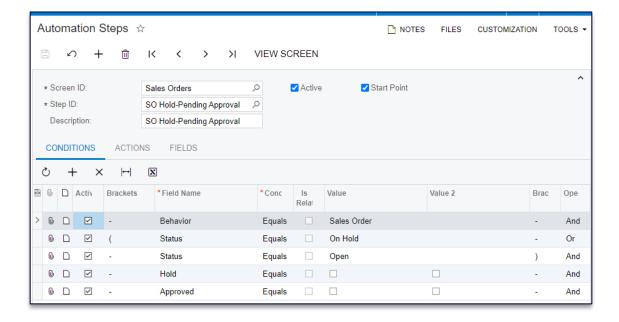
Preferences	Change workflow by selecting options in preference screens.
Assignment and Approval Maps	 Create approval and assignment maps with tools in the user interface.
State Workflow Engine	 Modify workflows, buttons, valid values, dialog boxes, and behaviors by modifying screen states in a customization project.
Code	 Write code to extend graphs to create custom workflows as described in <u>Help</u> using graph extensions.



Old Workflow (Automation Engine)

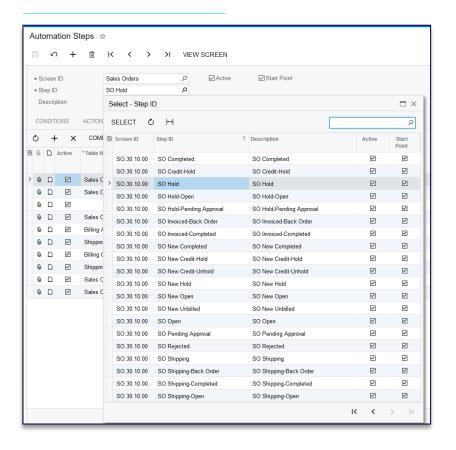
Document life-cycle management

No-code





Why Replace the Acumatica Automation Engine?





Not an obvious mechanism and structure



Not possible to make changes from code



Automation customization is not recommended



Hard to upgrade in case of local changes



Obscure inter-screen interactions



State Workflow Engine

Key Components

Screens

Workflows

States

- Fields
- Event Handlers
- Actions
 - Dialog boxes

Transitions

- Triggers
- Conditions

→ SCREENS ▼ PO302000 Actions (3, inherited 13) Event Handlers Fields (1) Conditions → Workflows (1, inherited 3) RT workflow RN workflow RX workflow Inspect Receipt Dialog Boxes (1)





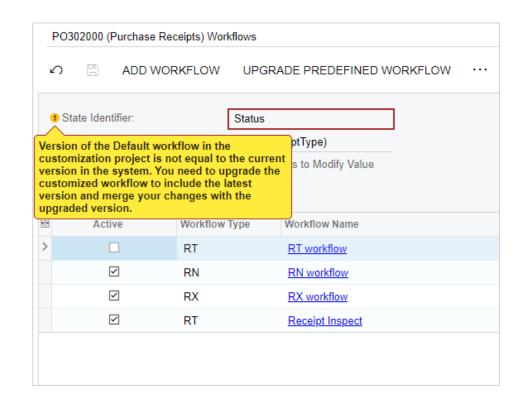
Workflow Basics

Default Workflows

Included with application

Modifying Workflows

- Extend workflows (make changes on top of default)
- Create new workflows





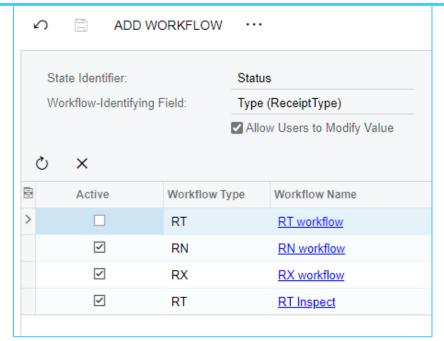
State Workflow Engine

Data Entry Screen Workflow 1

Workflow 2

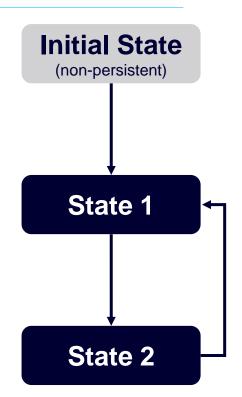
Workflow n

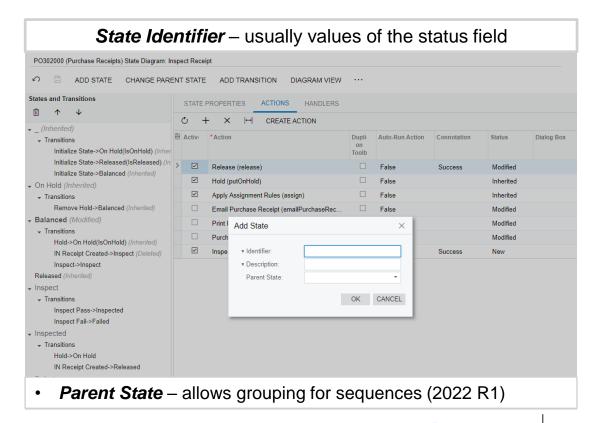
Workflow-Identifying Field – often a Type or Class field





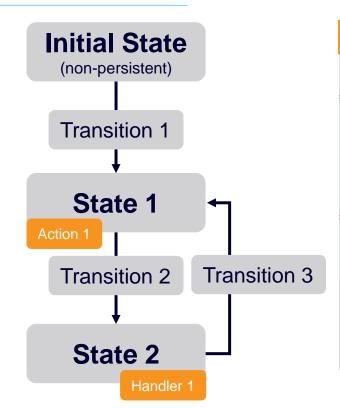
Workflow Structure: States

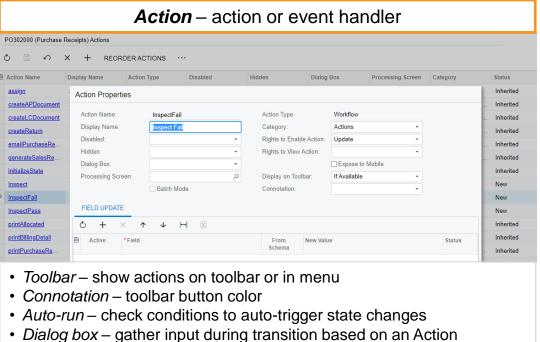






Workflow Structure: Actions





Workflow is Defined in Code

Framework Library: PX.Data.WorkflowAPI

Methods: Add, remove, replace, update

Workflow objects:

Forms, Conditions, Actions, Fields, Flows (fields, actions, transitions)

Process:

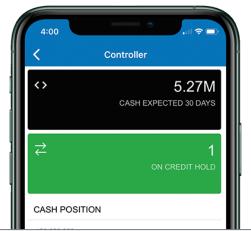
See example: Add Assignment to Opportunities

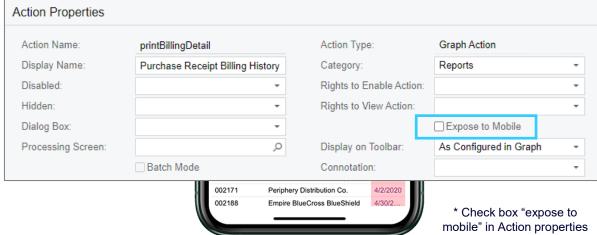
```
context.AddScreenConfigurationFor(screen -=>
    .StateIdentifierIs<status>()
    .AddDefaultFlow(flow·=>
        .WithFlowStates(fss -=>
            fss.Add(initialState, flowState => flowState.IsInitial(g => g.initializeState));
            fss.Add<State.hold>(flowState => ...);
            fss.Add<State.open>(flowState =>...);
            fss.Add<State.confirmed>(flowState => ...);
            fss.Add<State.partiallyInvoiced>(flowState =>...);
            fss.Add<State.invoiced>(flowState =>...);
            fss.Add<State.completed>(flowState =>...);
        .WithTransitions(transitions =>
            transitions.AddGroupFrom(initialState, ts:=>...);
            transitions.AddGroupFrom<State.hold>(ts·=>...);
            transitions.AddGroupFrom<State.open>(ts-=>
                ts.Add(t·=>·t.To<State.hold>().IsTriggeredOn(g·=>·g.putOnHold).WithFieldAssignment
               ts.Add(t => t.To<State.confirmed>().IsTriggeredOn(g => g.confirmShipmentAction).Wh
            transitions.AddGroupFrom<State.confirmed>(ts -=>
                ts.Add(t\cdot=>\cdot t.To<State.open>().IsTriggeredOn(g\cdot=>\cdot g.correctShipmentAction));
                ts.Add(t => t.To<State.invoiced>().IsTriggeredOn(g => g.OnInvoiceLinkedHandler).Wh
                ts.Add(t => t.To<State.partiallyInvoiced>().IsTriggeredOn(g => g.OnInvoiceLinkedHa
            transitions.AddGroupFrom<State.partiallyInvoiced>(ts =>...);
            transitions.AddGroupFrom<State.invoiced>(ts =>...);
        transitions.AddGroupFrom<State.completed>(ts =>...);
```

Workflow on Mobile

Mobile Workflow

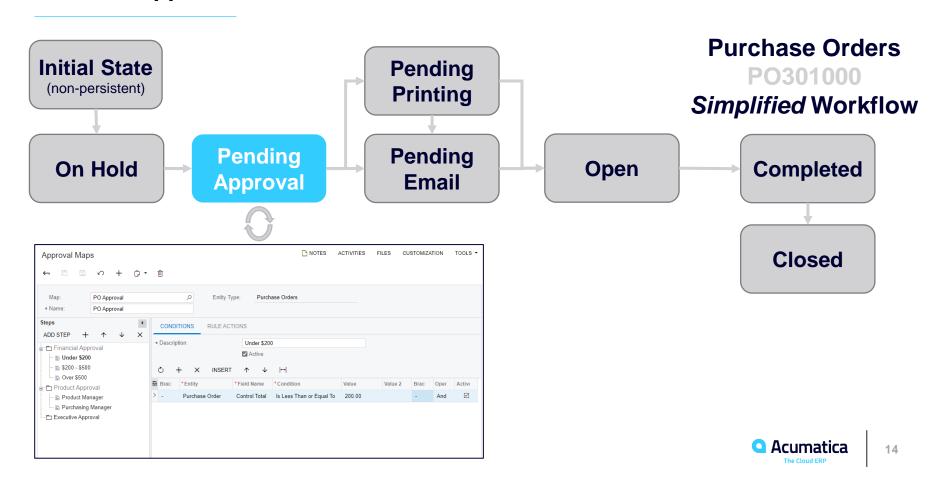
- Field changes
- Additional states
- Transitions
- Actions
- Event Handlers







How do Approvals Fit into Workflow?



Migration to the Workflow Engine

2020 R1

- CRM: opportunities, leads, cases
- Projects: subcontracts

2020 R2

- Distribution: sales orders, shipments, invoices
- Projects: request for information, project issue, drawing logs

2021 R1

Automation steps eliminated, except a few screens

2022 R1

- Add composite / parent states
- Financial screens updated to take advantage of engine

2022 R2

Stay tuned



Demonstration



Demo: Inspect Purchase Receipts

Scenario

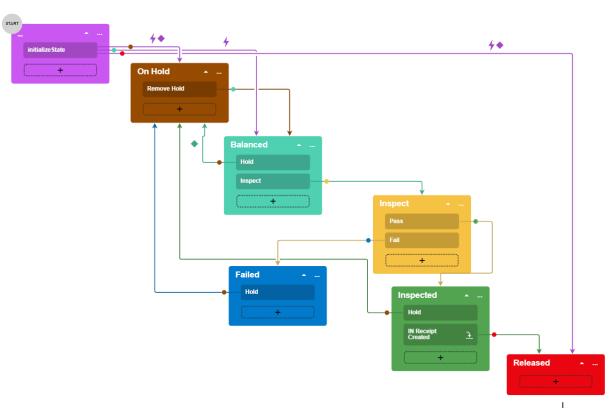
- When receiving a purchase order, inspect the order before the purchase receipt is processed and goods are added to inventory.
- If inspection is successful, then place in area for counting and releasing.
 - Save inspection results in note field
- If inspection fails, then place in quarantine area.
 - Allow to repair and re-inspect
 - Allow to ship back to vendor



Demo: Inspect Purchase Receipts

New flow diagram ...

- Add 3 states
 - Add 3 field values
- Add 3 actions
- Add/Modify 6 transitions
- Delete 1 transition
- Add 1 dialog box



Financial Scenario - Build in Code

Scenario:

Modify workflow for Invoices and Memo screen

Scenario Details:

- Modify order of existing composite/parent steps (pending email, pending print)
- Add additional state (pending review)
- Trigger events by workflow on field changes (subscribe on changing amount field trigger back to pending review state)

Benefits of Code

- Composite states easier to manage
- List of features only available by coding (for example: trigger events from workflow)

Customer Examples



Custom Workflow: Sales Orders

Scenarios from the Real World

- Added user defined fields on Sales Orders for use in customs reports
- Field values become available to customer only when Sales Order status is Shipping or Completed
- Used workflow to enable editing of UDF fields when status changed



Custom Workflow : Appointments

Scenarios from the Real World

- Field Service Appointments used for billing
- After appointment is closed, non-billable follow up might be required. The results need to appear on the invoice.
- Using workflow, notes are editable on completed appointments describing follow-up work
- Invoice is re-sent to clients with the additional notes



Custom Workflow: Cases & Payments

Scenarios from the Real World

- Cases used for professional service projects and billing
- Different workflows, actions, and reason codes needed depending on case class
- Implemented custom field for follow up
- **Payments** screen add additional reports to the action menu

Processing

Void

Inquiries

Reports

Corrections

Customer Details

Card Processing

AR Register Detailed

Print Receipt



Custom Workflow: CRM

Scenarios from the Real World

Rainbow Treecare

 Leads required changing lead status values to currently used terms "cold" and "warm"

ReproTech

- Opportunities require different processing and reporting for pick-ups and drop-offs
- Implementation:
 - Added different flows depending on the opportunity ClassID



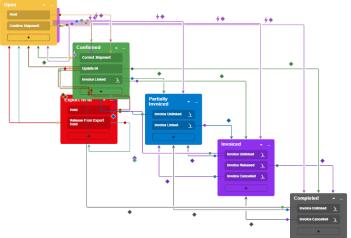
Custom Workflow: Sales Processing

Scenarios from the Real World

PICS Telecom

- Sales of recycled telecom parts and equipment require processing by an export compliance group.
- Sales Orders contain user defined fields to capture export information.
- Shipments include an additional state called Export Hold which must be reviewed by the compliance team









What's New and What's Coming



New and Planned Workflow Enhancements

New in 2022 R1

- Change field values when entering or leaving a state
- Composite states to simplify building workflows
- Financial Screens support workflow changes

Coming Soon & Future Ideas*

- Set drop down field value from target states (2022 R2)
- Conditional display for dialog boxes (2022 R2)
- Set default values for non-primary elements (2022 R2)
- Trigger business events from the Workflow Engine (2023 R1)
- Add parent states to diagram view





^{*} All future dates subject to change

Summary and Key Points

- Workflow defined in Acumatica can be easily customized by both a developer using code and by an end user from workflow designer.
- Use Workflow to define customizable business logic.
- Workflow mechanism is a plain old finite state machine.
- Info and practice:
 - Open University Course <u>W150</u>
 - Code example: <u>Add Assignment to Opportunities</u>







Thank You!

www.acumatica.com/developers