Workflow Implementation for

Event Management

Table of Contents

[Introduction 4](#_Toc410991013)

[Event Management Scenario 4](#_Toc410991014)

[Visual representation 5](#_Toc410991015)

[Workflow Category 7](#_Toc410991016)

[Query 7](#_Toc410991017)

[Workflow type 7](#_Toc410991018)

[Workflow approval tasks 10](#_Toc410991019)

[Workflow automated task 12](#_Toc410991020)

[Adding supported elements in workflow type 13](#_Toc410991021)

[Enable Workflows on Forms 14](#_Toc410991022)

[Override submitToWorkflowMethod 15](#_Toc410991023)

[Submit to workflow class 16](#_Toc410991024)

[Event Handlers 16](#_Toc410991025)

[Authoring the workflows 17](#_Toc410991026)

[Workflow configuration 18](#_Toc410991027)

[Workflow editor 19](#_Toc410991028)

[Dynamic assignment of user to invitation response approval task 20](#_Toc410991029)

[Custom Workflow participant provider 21](#_Toc410991030)

[Participant provider assignment 22](#_Toc410991031)

[Workflow enabled forms 25](#_Toc410991032)

[Workflows on Enterprise Portal 26](#_Toc410991033)

[Enabling event approval on Enterprise portal 26](#_Toc410991034)

[Enabling workflow type 27](#_Toc410991035)

[Enabling workflow element 28](#_Toc410991036)

[Enterprise portal with Worklfows 29](#_Toc410991037)

[Things to keep in mind 29](#_Toc410991038)

Highlighted Features

|  |  |
| --- | --- |
| Features | Purpose |
| Workflow types | Two workflow types: for event approval and for invitation |
| Workflow Approval Items | One approval item for each approval and invitation |
| Workflow Automated Task | To send invitations automatically once event is approved |
| Workflow Participant Provider | New Class created to dynamically assign users to an approval task for event invitations |
| Workflow Alerts | Configuring alerts within AX and email |
| EP Enabled WOrklfow Items | Customization to enable workflow on enterprise portal |

Workflow Overview

## Introduction

Workflow is a system in Microsoft Dynamics® AX 2012 that allows business processes to be automated. For example, a purchase requisition may need to be approved by a number of different employees according to the requisition's total amount. Each employee has to approve it before the next employee in the approval route.

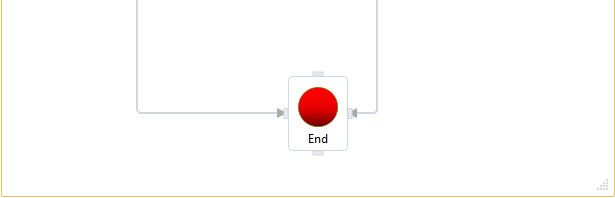
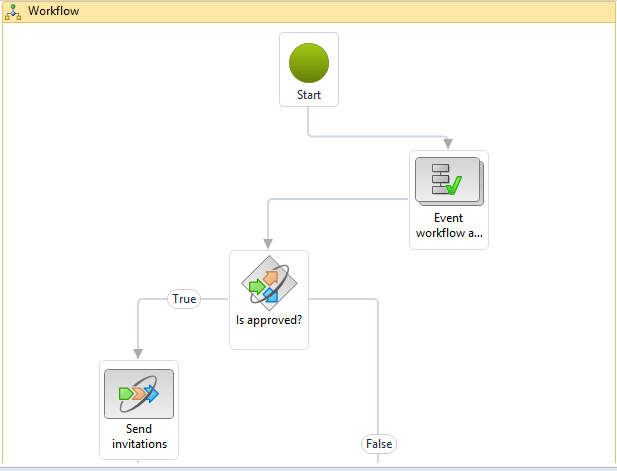
A Workflow in Microsoft Dynamics AX uses a combination of Application Object Tree (AOT) elements created by IT and additional set up that a user can control.

## Event Management Scenario

Whenever an event is drafted and ready to be launched, it needs to be approved by the concerned authority before publishing. Once the event is approved, an automated task should run which will send separate workflow to each invitee as an event invitation. The requirement is whenever an event is created, it should undergo the approval. The assigned user can either approve or reject the event. Until it is approved, the event cannot be published. After the approval, each workflow invitation should be handled as approval.

***Note: This document doesn’t provide the step by step processes and assumes the reader has the basic understanding of workflow development and customization. This document states the implementation of workflow for event management. Purpose of this document is to provide the details of live issues and requirements came across for event management and how they are resolved using workflows. Please refer to the Chapter 6 of Development 3 learning material for Microsoft Dynamics AX 2012 for the step by step process and basic workflow development.***

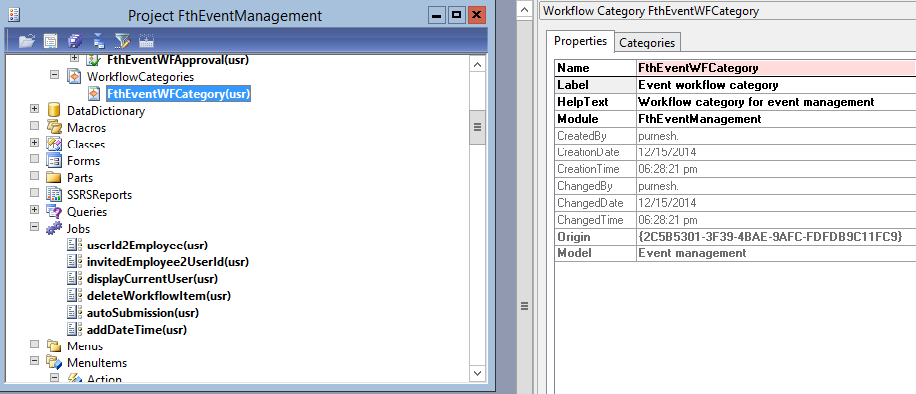
## Visual representation



Workflow Development

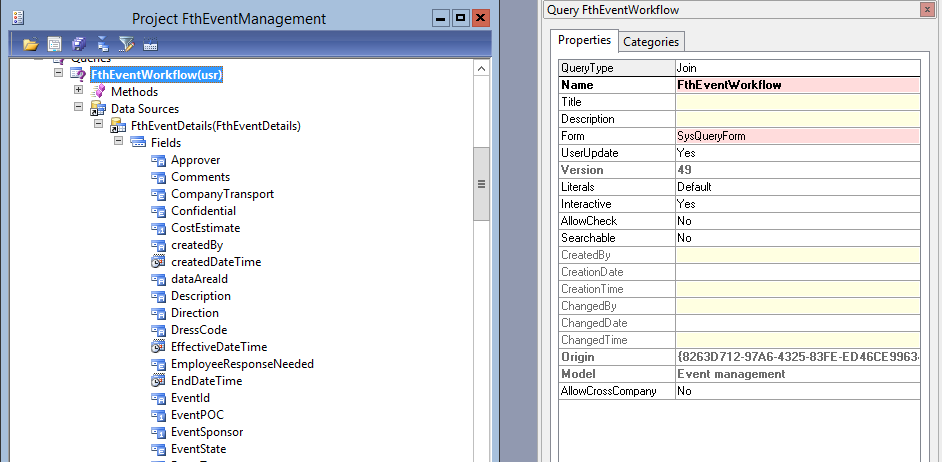
## Workflow Category

Workflow category is used to determine the module in which the workflow will be available. Modules are defined by the ModuleAxapta enum. We created workflow category for Event Management Module.



## Query

A workflow uses a query to define the data that is used by the workflow. It can define one or more tables and all or selected fields on that table. We create query for Event details table.



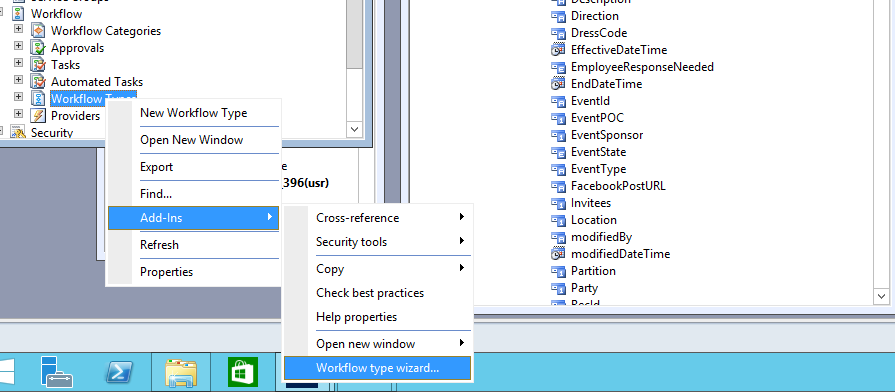
## Workflow type

A workflow type defines the business document being acted upon, the supported AOT elements, and event handlers and menu items for the workflow. Workflows are created based on a type, and many workflows can be based on the same type.

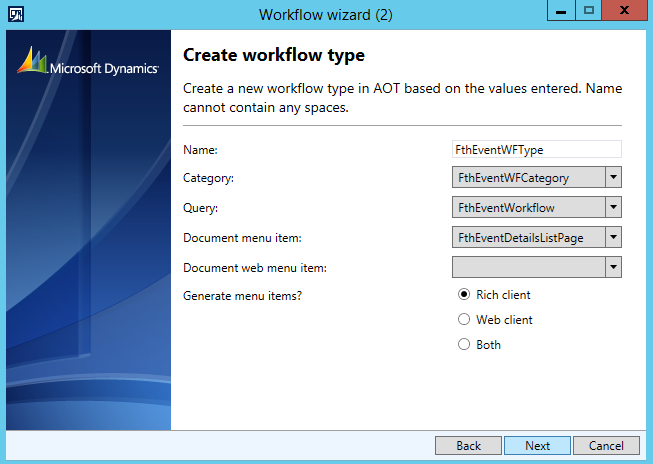
For event management, we have created two workflow types:

1. For Event approval – This is required for event approval and sending invitation workflows
2. For Event invitations – This will be a separate workflow for each invitee.

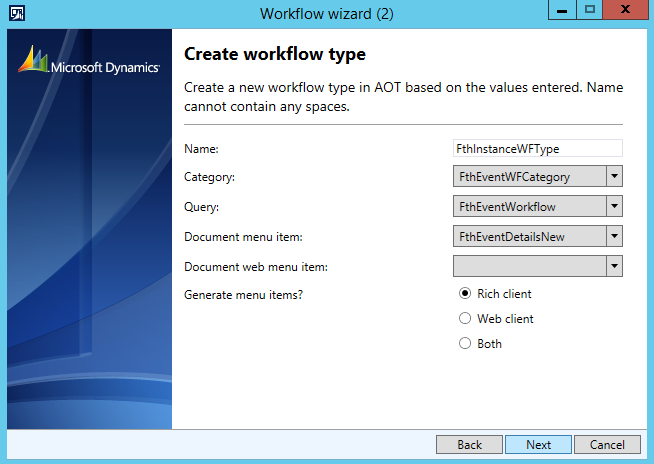
Workflow types are created using the workflow type wizard.



Workflow type wizard



Event approval workflow type



Event invitation workflow type

## Workflow approval tasks

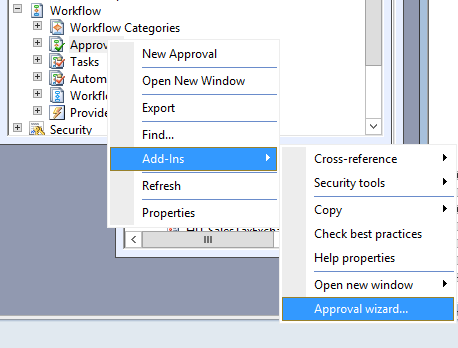
A workflow element may contain a number of outcomes. The workflow may be approved, rejected, returned or a change may be requested. The Workflow Approval element determines which of these outcomes is allowed and what happens in the event of each outcome.

Each outcome can trigger the execution of business logic by specifying a menu item for each item.

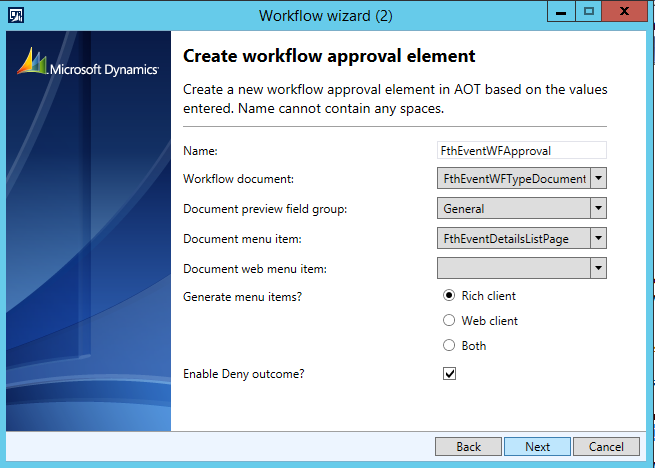
For event management again, two approval tasks are required:

1. Event approval – This will be the part of event approval workflow type responsible for event approval
2. Event invitation – This will be the part of event invitation workflow type responsible acceptance/rejection of event invitation.

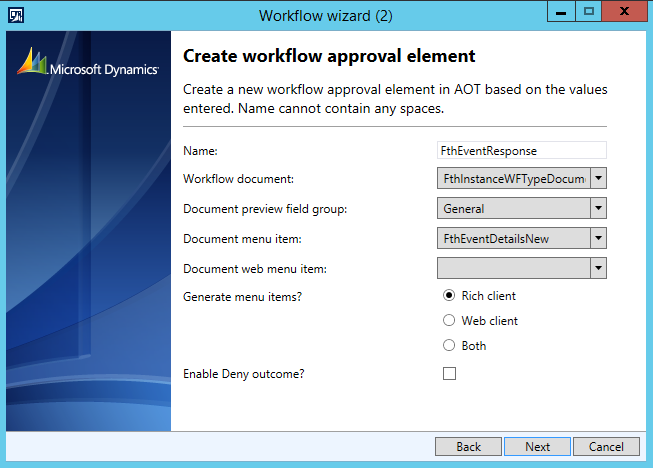
These workflow approval tasks are also created by wizard:



Approval wizard



Event approval task



Event invitation response approval task

## Workflow automated task

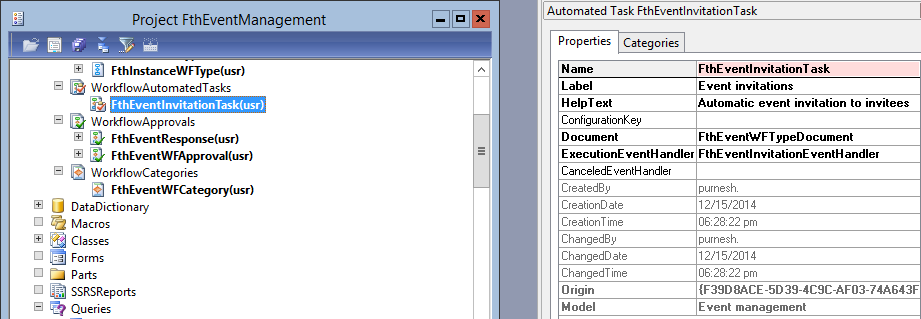
Automated tasks are workflow elements that are used to start X++ code in the application without requiring user intervention. When the workflow is started, an automated task could execute a customer credit check and send the results to the workflow owner. Based on the results of the credit check, the workflow owner can take action and continue the sales order workflow.

The business logic for an automated task must always run on the AOS and not the client. You can use an automated task to, for example, execute database transactions or updates.

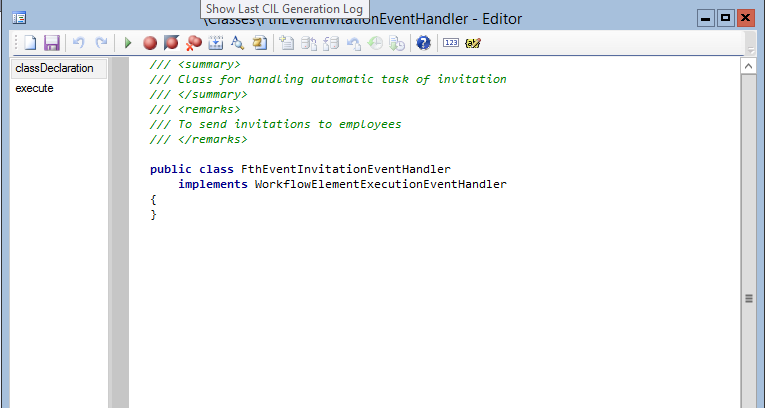
Automated tasks do not have outcomes. Since there is no user interaction, the outcome of an automated task is one of the following: completed, not completed, or canceled. You can write custom business logic enclosed in event handlers to handle these outcomes.

For event management, we have created one automated task – send invitations for executing the code which sends invitations to the invitees. It points to a class which acts as the event handler for the task and executes two tasks:

1. Instantiate event invitation workflows for each invitee.
2. Sends outlook calendar invites for each invitee.



Event invitation automated task

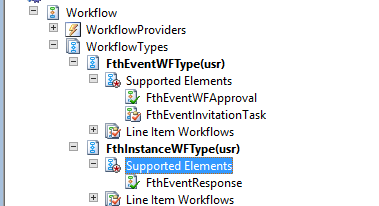


Event handler class with execute method containing business logic

## Adding supported elements in workflow type

In order to use the approval tasks or automated tasks etc. in your workflow configuration you must add these tasks to their respective workflow types under supported elements.

For event management, event approval task and event invitation automated task is put under event approval workflow type whereas event invitation response approval task is put under event invitation workflow type.



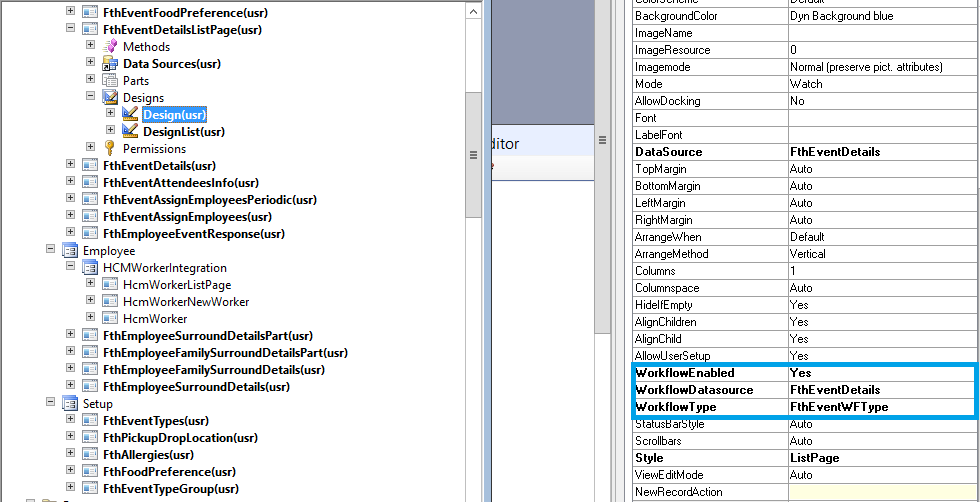
Supported Elements for Workflow types

## Enable Workflows on Forms

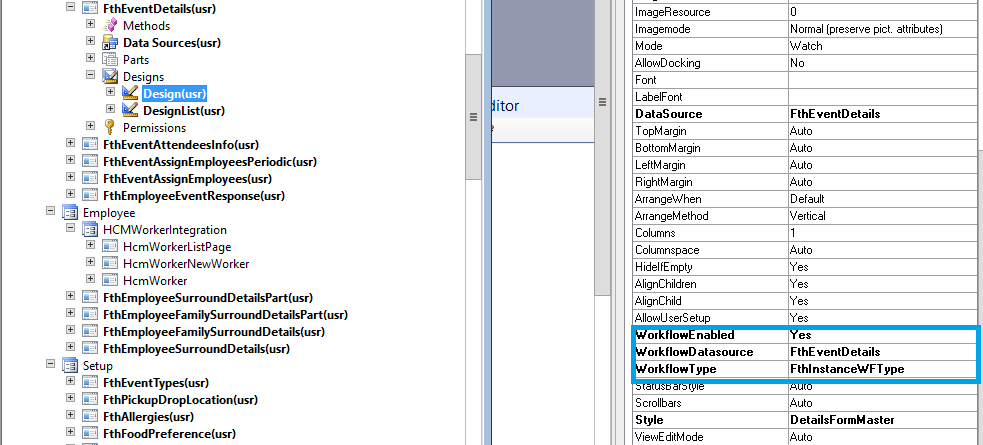
Now that both the workflow type and approval element are defined, you can specify which forms will use it.

Workflow on the form is enabled using properties on the design node, and by overriding a form method

For event management, event approval workflow is enabled on event details list page and event invitation response workflow is enabled on Event details master form. However, event invitation response workflow will be enabled only when the event is published.



Event details list page



Event details master form

### Override submitToWorkflowMethod

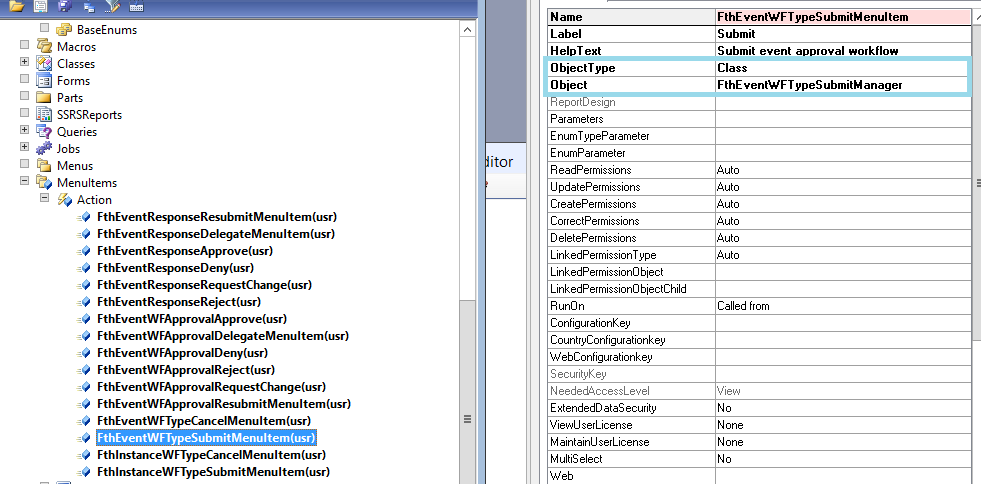
For event management, we have created method in event details table. It is this method which contain the logic for allowing submit or not. This method returns a Boolean value. If true, submit will be enable else not.

For example, in case of event management we are checking on the basis of event states. For event approval workflow submit will be enabled if the event is in draft state else not.

### Submit to workflow class

To submit a document for workflow processing, call standard code to prompt the user for a comment and to process the submission. When you create the workflow type using wizard, a private project will be create and inside there will be one submit manager class, so the submit menu item will point this class. In the main method of this class you must right the logic what you want to do after submission.

For event management, we are updating the event state from draft to submitted.



Submit menu item for event approval workflow type

## Event Handlers

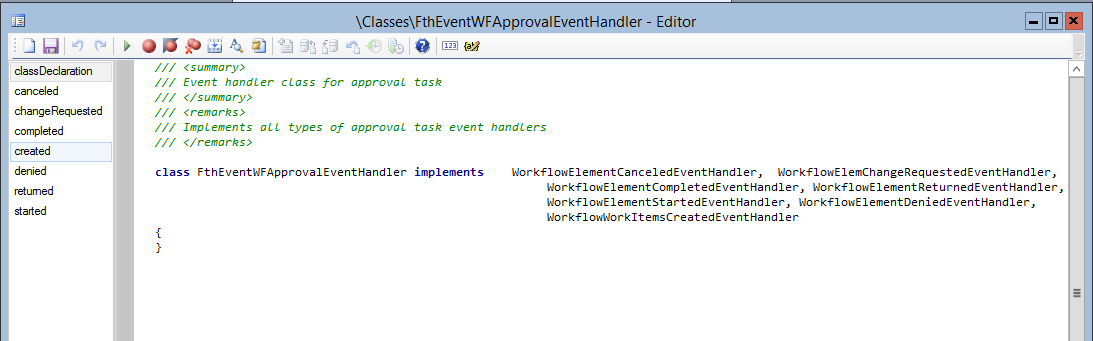
Event handlers are used to execute business logic at specific points in the life of a workflow. They can be implemented at the workflow level, for example when the workflow is started or completed, or at an element level, for example when anyone approves or rejects a step in the approval.

Event handlers are implemented by creating a class that implements one or more of the EventHandler interfaces. Wizard will automatically create the classes and event handlers for you. You just have to write your business logic in one or more of those methods as per your requirement.

There are two types of event handlers

1. Event handler for the entire workflow
2. Element level event handler

For event management, we are using the element level event handler of event approval task, using completed event method to update the event state to approved from submitted. So if the approver approves the event completed method will execute and change the event state.



Event approval task event handler class

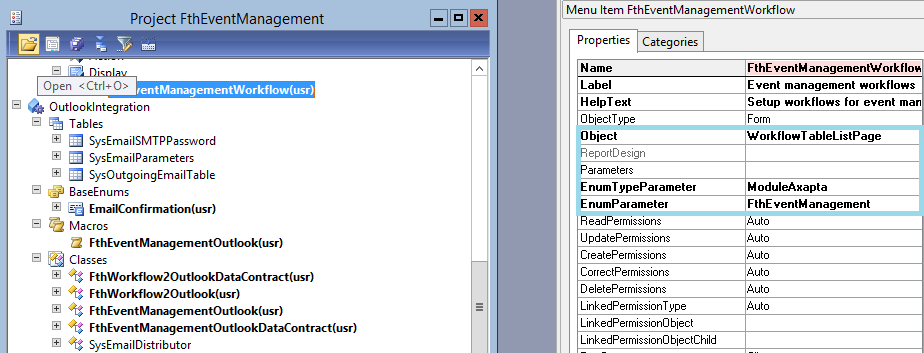
***NOTE****: Code run in batch jobs must first be compiled to the Common Intermediate Language (CIL) of the .NET framework. This increases the performance of the batch job.*

## Authoring the workflows

Now that you have created the workflow types and enabled it on forms, you can configure it for use. First thing you need to put your workflow category in the main menu. So you would have to create a display menu item and point it your module.

For event management, we have created event management workflows menu item and have placed it under the setup submenu.

*Note: Your module should be present in the ModuleAxapta enum and main menu.*

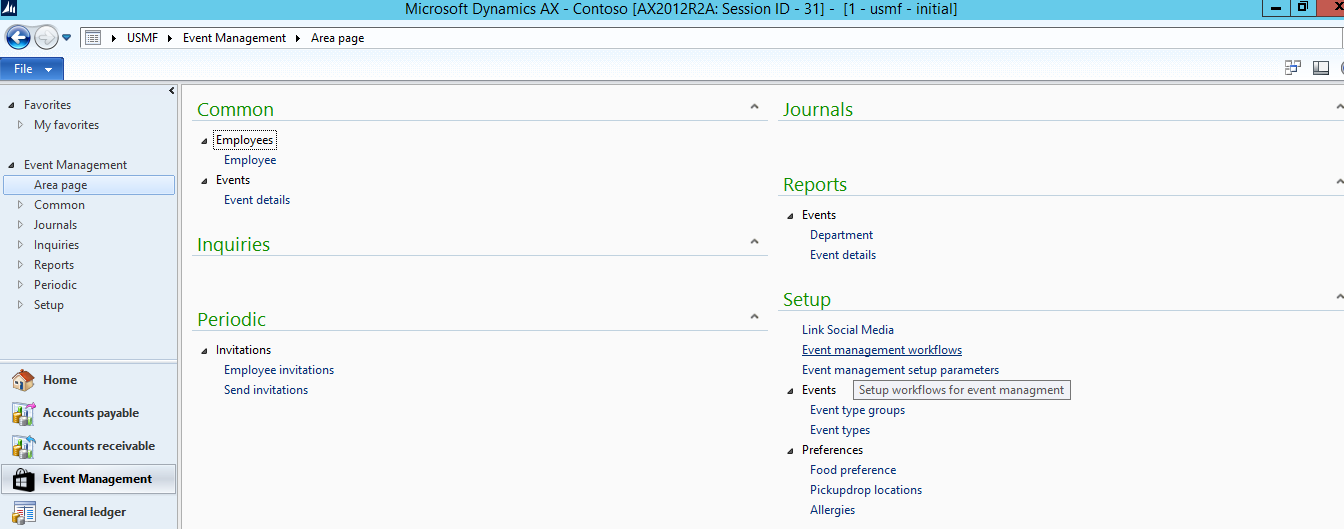


Event management workflows menu item

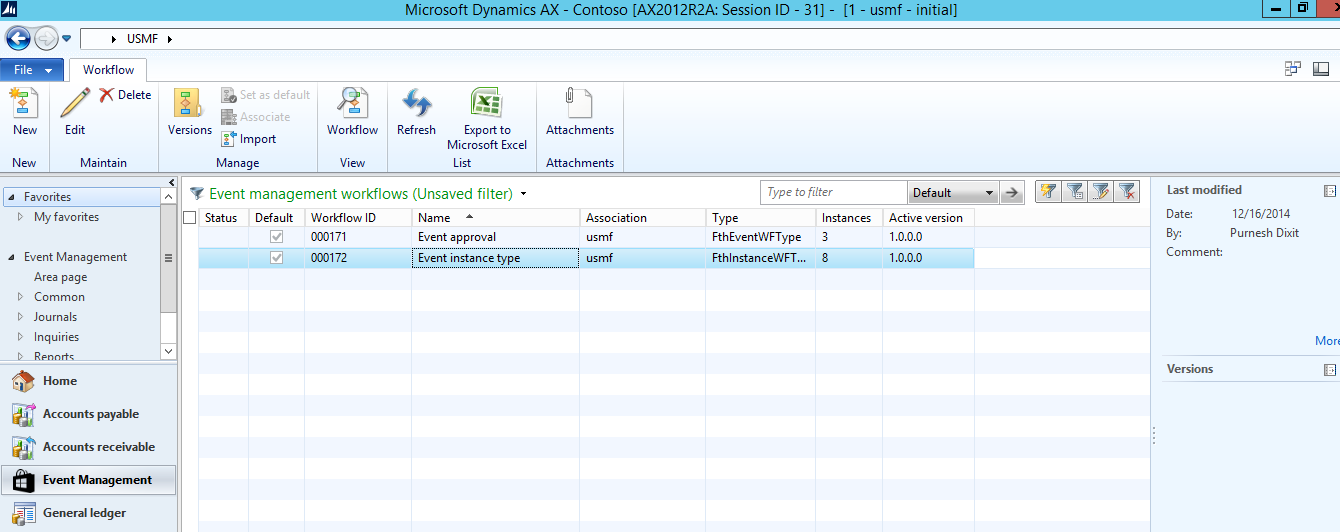
### Workflow configuration

Now that you have placed your menu item on the client, it’s time to configure it based on your requirement.

For event management, the workflow client configuration can be accessed under setup submenu.



Event management module

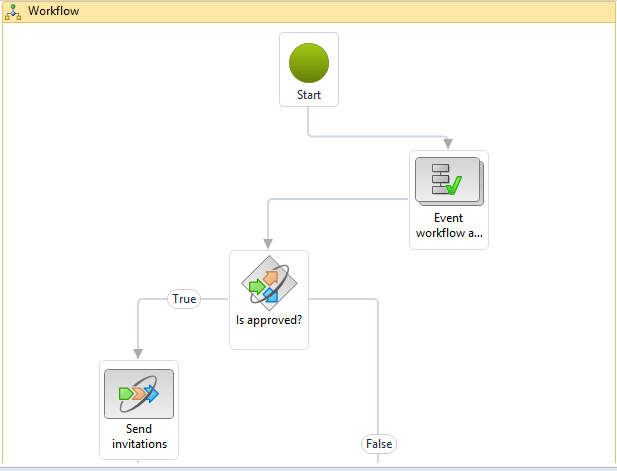


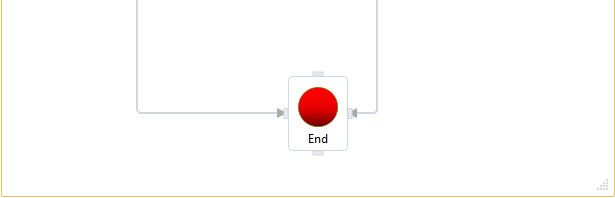
Event management workflows

These workflows can be configured in the workflow editor very easily. Let us have a look at event management workflow editor.

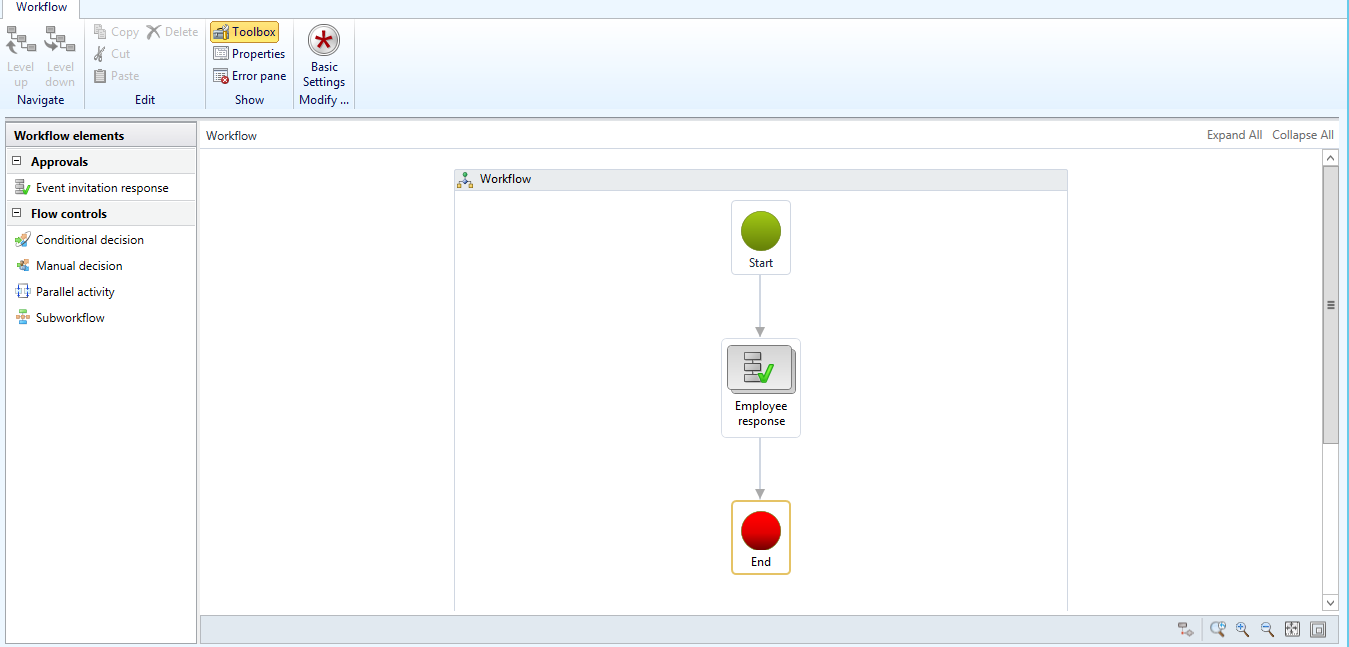
### Workflow editor

This is a GUI editor, where you can place your workflow elements and design a flowchart for your configuration. For each elements there are specific properties that you would have to set based on your requirements. For event management, we have to workflows as mentioned already, so we have two workflow configurations.





Event approval workflow configuration



Event invitation response workflow configuration

As the scenario states the requirements, these workflows are configured to serve the purpose. With the GUI interface the flow of the workflows are easily understood.

The development done by us as mentioned in previous parts, combines with the configuration and executes the tasks such that all the requirements are met optimally.

Once the configuration is completed, the workflow becomes enabled on the respective forms and is ready to be used.

## Dynamic assignment of user to invitation response approval task

For the event management, as we are using the workflow as sending invitations for events, there is a customization required as it cannot be achieved through standard functionality.

Requirement is that it is not fixed who will be invited so we cannot assign the task to particular user(s). We needed a way to dynamically send the workflows to only those people who are invited.

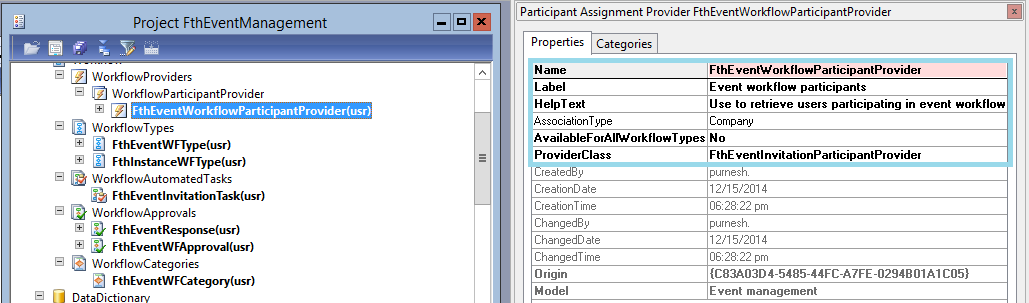
Thus, we needed to create a business logic which resolves the users based on the event invitees and assign them the invitation approval task.

For that we had to create our own Custom participant provider class for resolving the users.

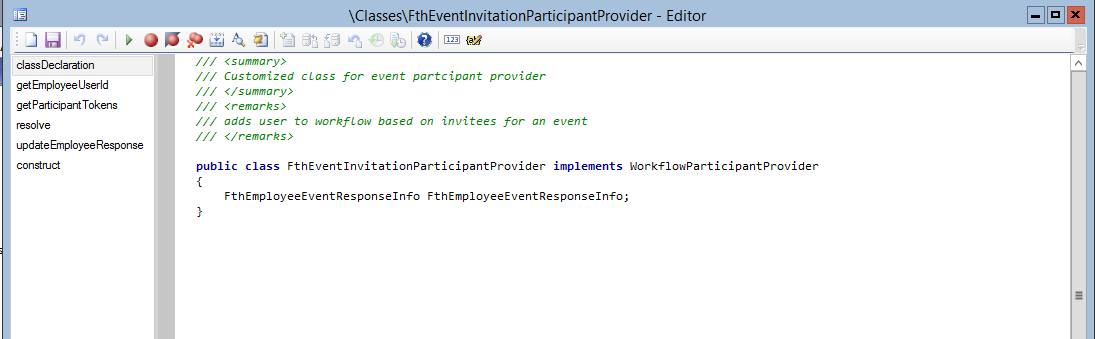
### Custom Workflow participant provider

Participant Providers in Dynamics AX are very useful classes for one to use in order to customize how work items are assigned within a workflow based on your specific business rules.

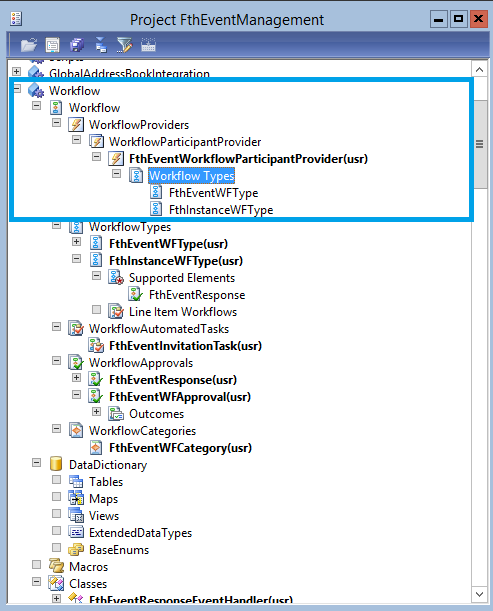
For event management, we created the event invitation participant provider to assign the work items to event invitees.



Event invitation participant provider



Event invitation participant provider class



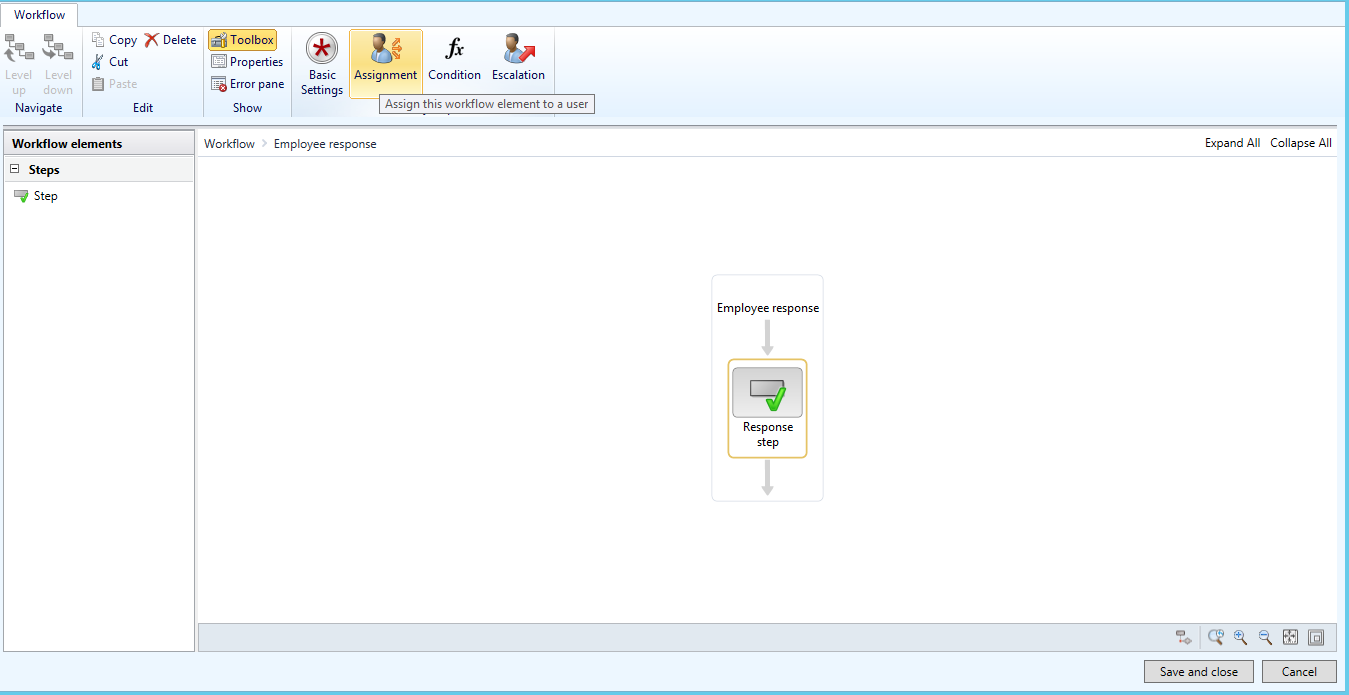
Adding workflow types to participant provider

### Participant provider assignment

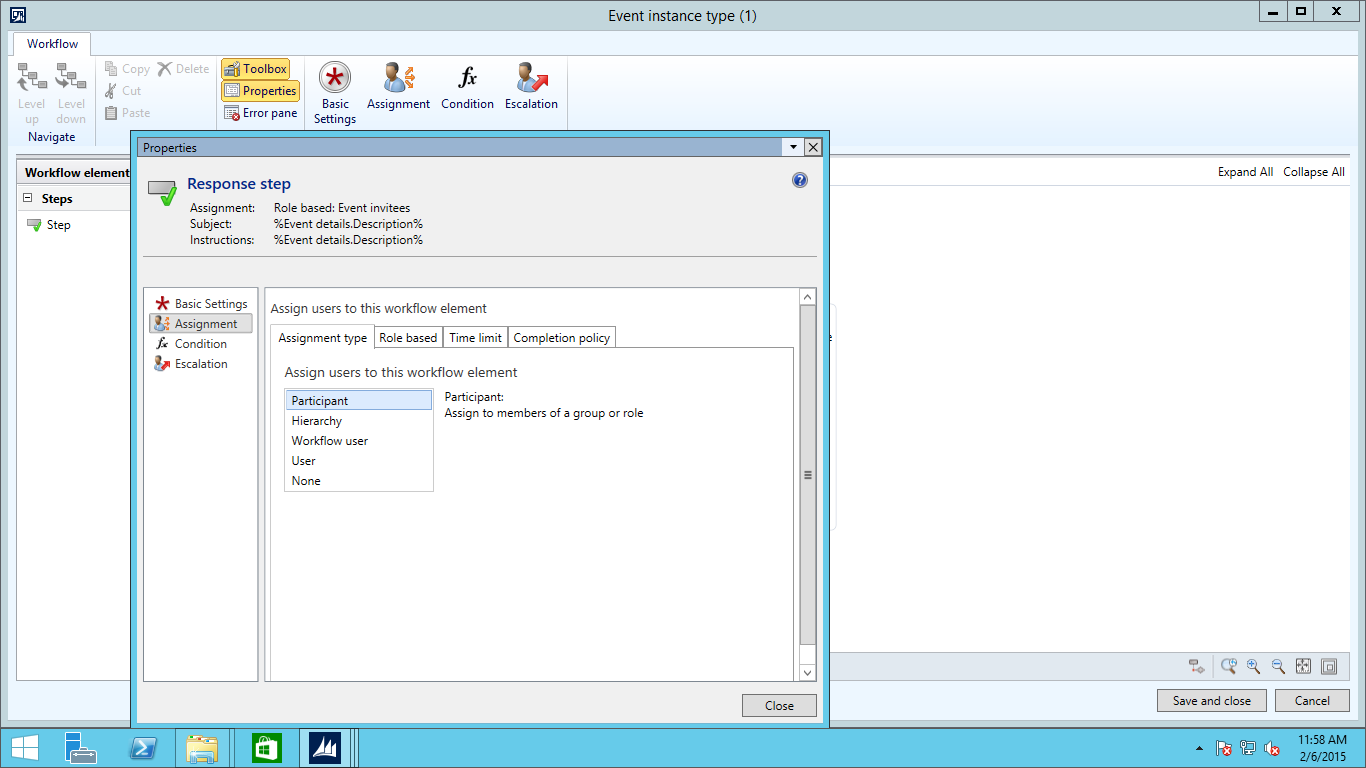
This class has some standard methods which needs to be overridden as per your business rules.

For event management, we have resolved the users who are invited which is done resolve method of the class.

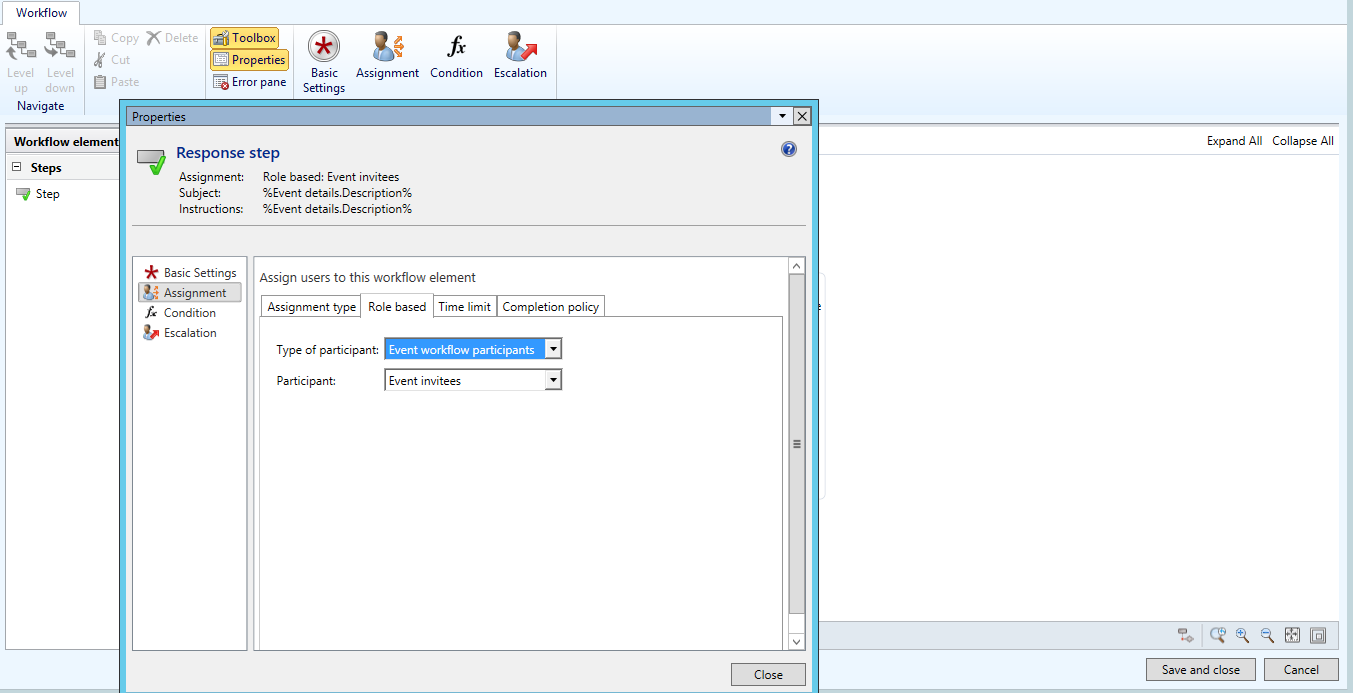
Now we need to assign this participant provider to the event invitation approval task.



Event invitation response step



Participant provider assignment

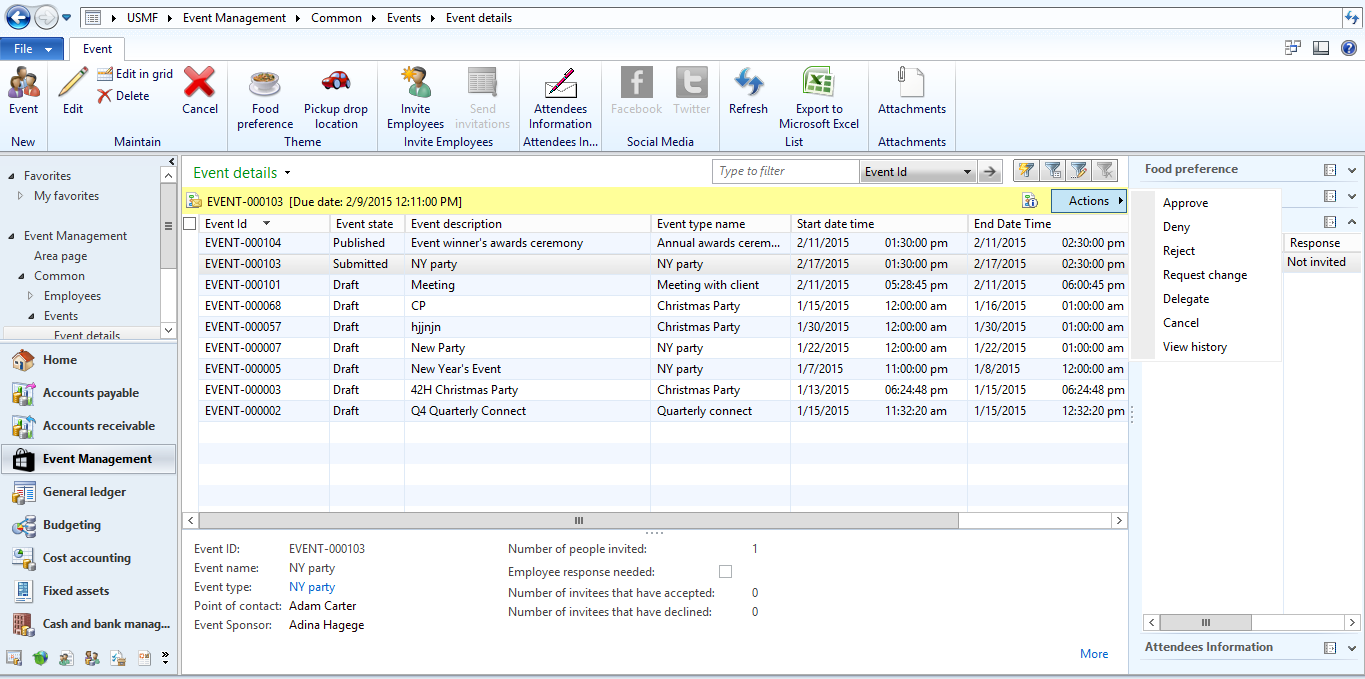


Role based assignment

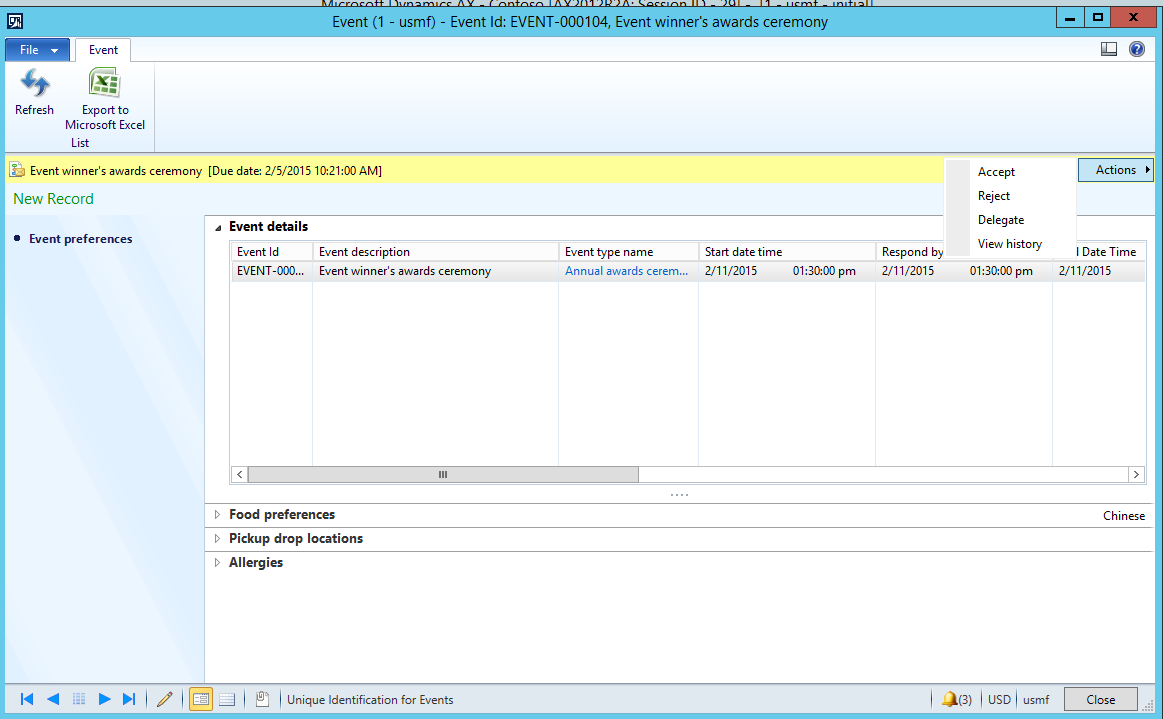
Now our workflow is capable of dynamically assigning work items to users. So for event management, after the event is approved, the workflow will be sent to only event invitees and the work Item will be assigned accordingly.

## Workflow enabled forms

Now the configuration and development is done, we can see the fully working workflows for event management forms.



Event approval



Event invitation response

## Workflows on Enterprise Portal

Microsoft Dynamics AX 2012 provides you a very useful feature of using the same workflows which you have customized on the Enterprise portal as well.

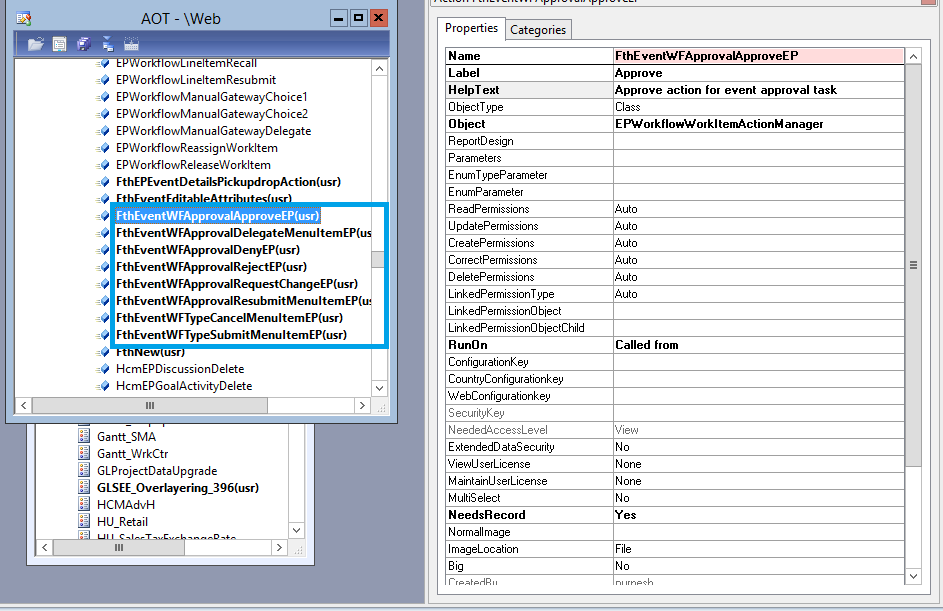
There are two ways you can achieve this:

1. Through the wizard while create workflow types and elements
2. Manually creating web menu items for the elements and workflow types.

For event management, we have followed the second way so as to customize exactly based on our requirements.

### Enabling event approval on Enterprise portal

In order to achieve this, first we have to create action web menu items for all the elements and workflow types, which want to see on enterprise portal.



Action web menu items for event approval

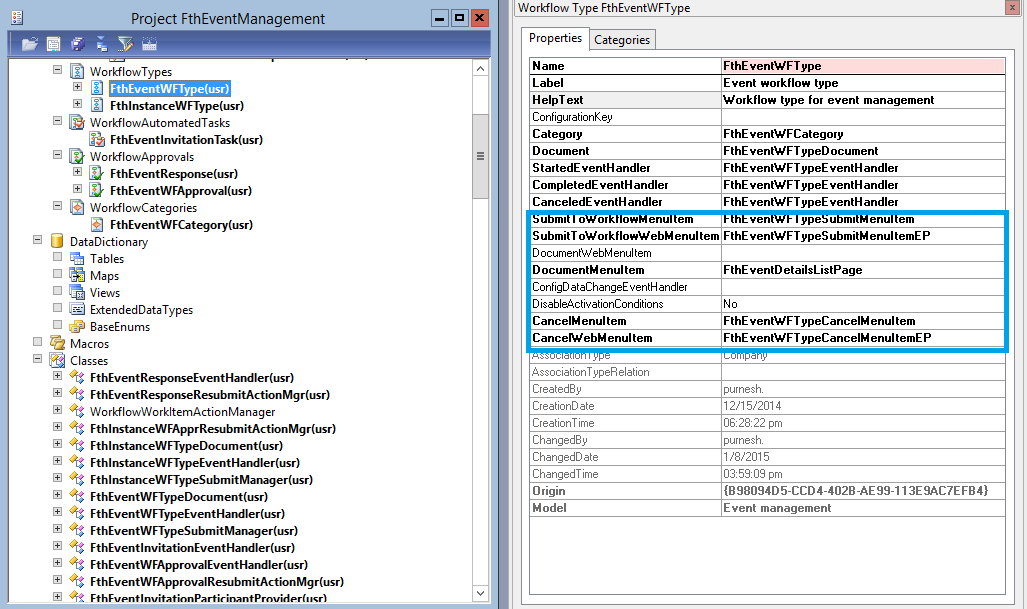
Two things have to be kept in mind

1. The object property has to be set to **ep classes of workflow.**
2. Needs record property should be **Yes, if you want to pass the record in args.**

### Enabling workflow type

After creating the action web menu items, we need to set web menu items to respective action web menu items for workflow types.

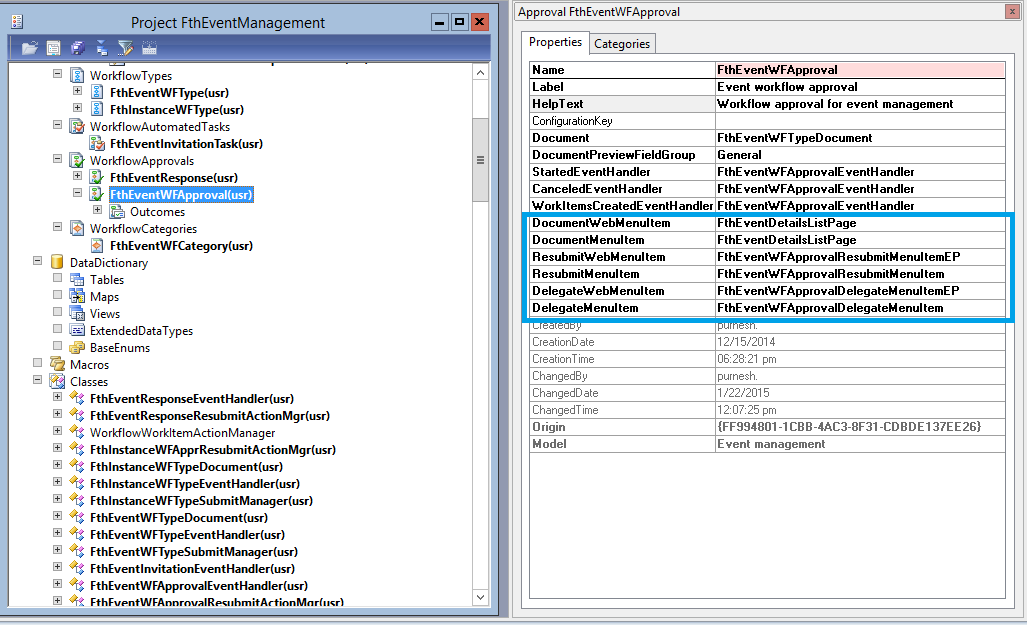
For event management, we have modified event approval workflow type.



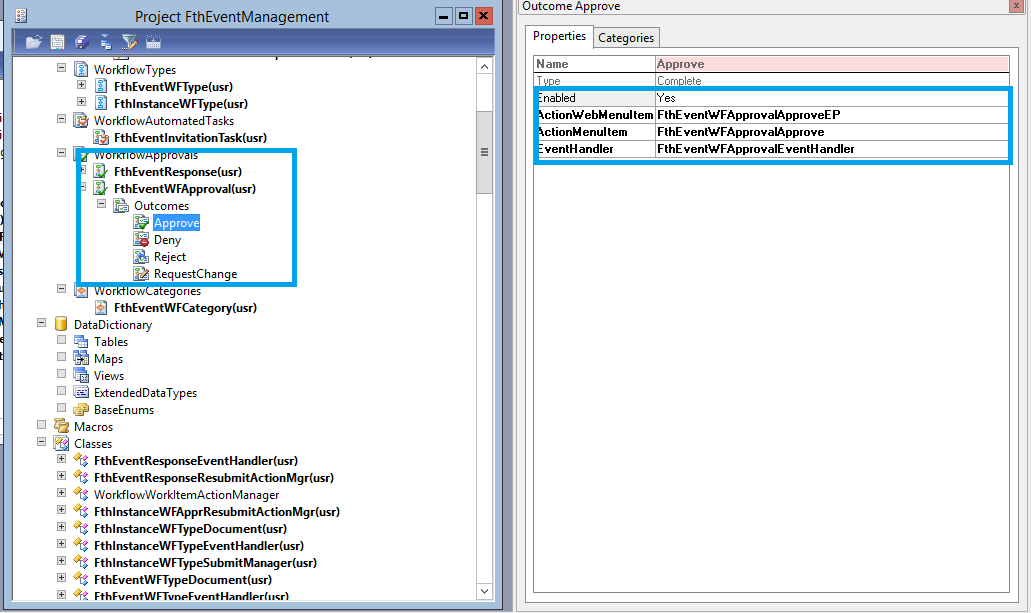
Event approval workflow type

### Enabling workflow element

Now we need to enable the approval workflow element for event management on enterprise portal. We did it by setting the web menu items of event approval element to their respective web action menu items.



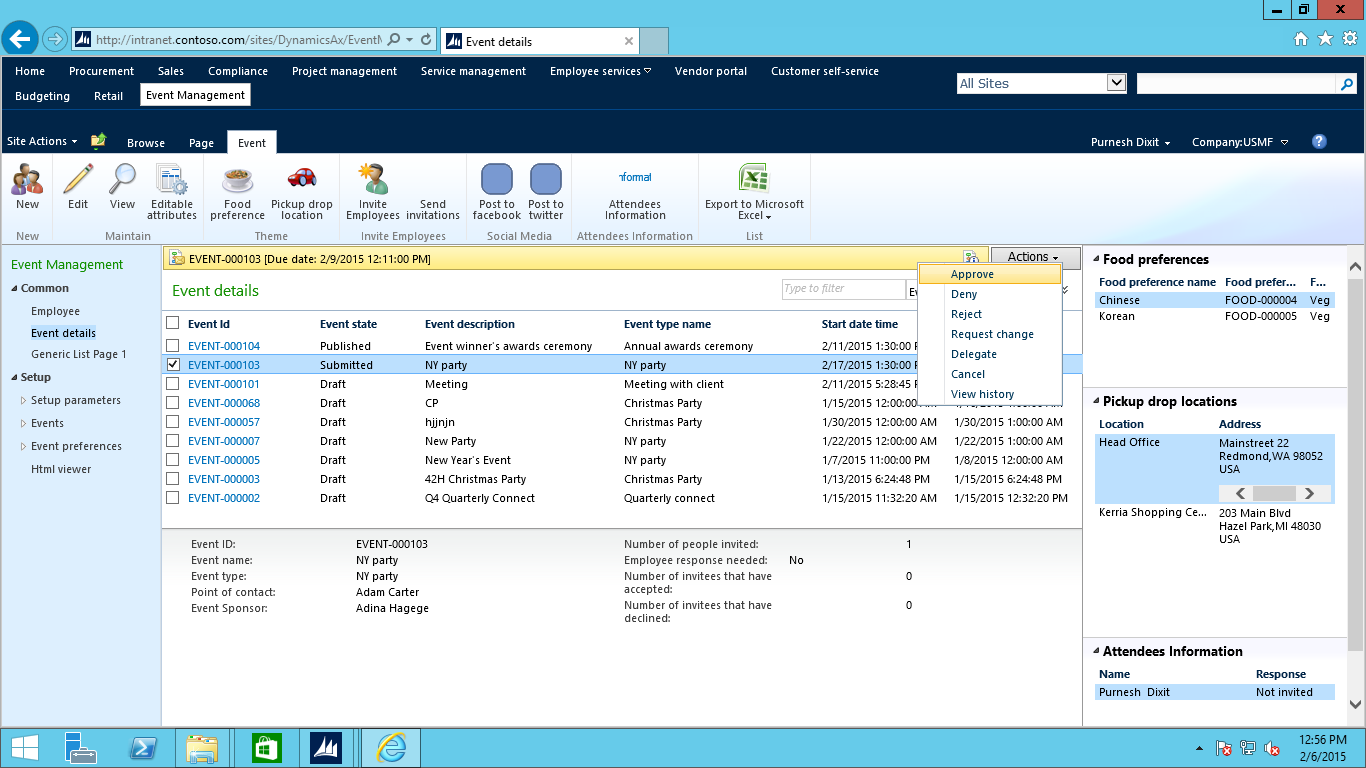
Event approval element



Event approval element outcomes

### Enterprise portal with Worklfows

Once you have enabled the workflow and their required elements on Enterprise portal, you can now see them on your respective EP forms.



## Things to keep in mind

There are few things which may come across very often so you need to always keep in mind

1. Always run Incremental IL if changing anything in code of workflow classes or related tables.
2. Always save the configuration with a different version if modifying anything with the configuration of workflows
3. See the workflow history for all the steps and errors.