NINTEX Workflow for Project Server

# Guide to Automating Workflows Quickly and Easily

Part 1

work flow for every one

NINTEX

# Table of Contents

# Introduction

Welcome to our Guide to *Automating Workflows Quickly and Easily: Nintex Workflow for Project Server 2010*. We will look at Nintex Workflow for Project Server 2010 and the ways it can empower your Project Managers to automate project, portfolio and collaboration workflows quickly and easily.

This guide is the first in a three-part series discussing each of the workflow types provided by Nintex Workflow for Project Server 2010 providing easy to follow step-by-step examples.

Nintex Workflow for Project Server makes the lives of Project and Portfolio Managers a lot easier, as they use a simple web-based editor to build and maintain workflows, removing the reliance on complex and costly development cycles. Nintex Workflow for Project Server is a graphical, web-based workflow designer built for the SharePoint and Project Server platforms. The key capabilities outlined in this guide are:

| | |
|---|---|
| **Project, Portfolio, and Collaboration Workflows** | Demand Management Workflows<br>Event Driven Workflows<br>Project Workspace Workflows |
| **Quick and Easy Functionality** | Browser Based Designer<br>No Code Required<br>Drag and Drop |
| **Connectivity with Other Systems** | Query Project Data<br>Connect to Internal and External Systems<br>Integrate with Office 365 |

Figure 1: The pillars of Nintex Workflow for Project Server

## Handling the Full Range of Project, Portfolio, and Collaboration Workflows

Project and Portfolio Managers have a range of different needs for automating tasks and processes. A workflow tool should fulfill the needs for designing processes related to demand management, event driven, and collaborative workflows.

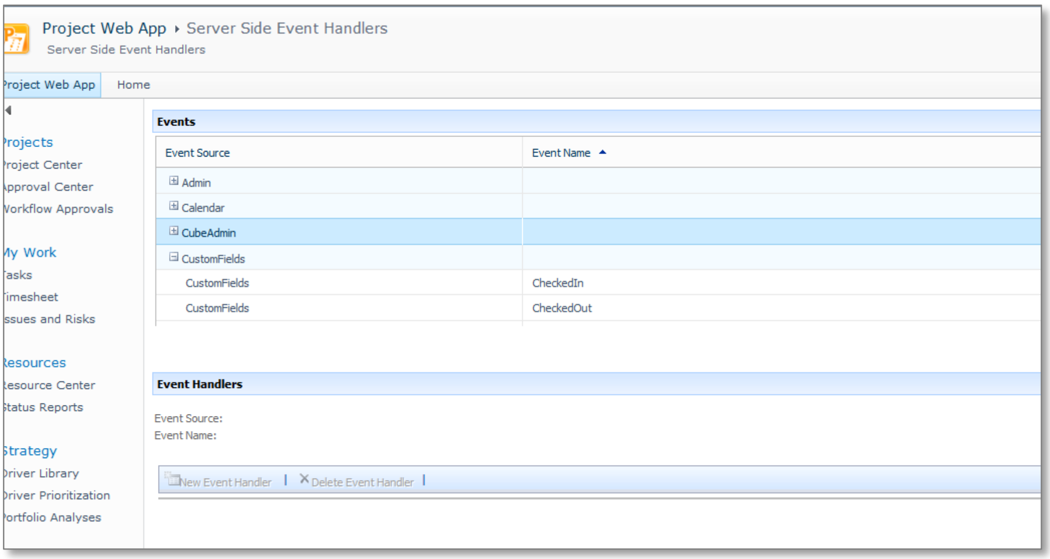
### Demand Management Workflows

Demand Management Workflows allow the orchestration of an organization's complete governance process using Project Servers demand management capabilities. Through these workflows, project ideas and projects can be guided through a governance process, whilst applying business rules, validations and approvals along the way.

### Event Driven Workflows

Nintex Workflow for Project Server introduced the ability to associate workflows with Project Servers event handling infrastructure. Project Server has the ability to associate bespoke code or assemblies against events that are raised by Project Server, such as Project Published, Project Saved or Custom Field checked in etc. With Nintex Workflow for Project Server 2010, it is now possible to associate a Nintex Workflow against a subset of events within the event handler infrastructure quickly and easily. Nintex Workflow for Project Server Event Driven Workflows allows you to create workflows that automate scenarios such as:

- Alert Project Managers when a resource is added or activated
- Notify management when a project is published
- Query line-of-business data when a project starts
- Sync custom fields and lookup tables across Project Web App sites
- Keep team members in the loop through integration with SharePoint and much more.



Events

| Event Source | Event Name ▲ |
|---|---|
| ⊞ Admin | |
| ⊞ Calendar | |
| ⊞ CubeAdmin | |
| ⊟ CustomFields | |
| CustomFields | CheckedIn |
| CustomFields | CheckedOut |

Event Handlers

Event Source:
Event Name:

Figure 2: Project Server Service Side Event Handlers

## Collaboration Workflows

Nintex Workflow for Project Server 2010 is built on top of Nintex Workflow 2010, and provides a number of capabilities that can be leveraged within the project workspaces; ranging from document approval workflows and notifications, to stakeholders when certain risks or issues are logged, through to synchronizing content from the workspace to an external Office 365 account for collaboration with external parties.

In addition, Nintex Workflow for Project Server also leverages a number of core capabilities from Nintex Workflow 2010, including:

- Leverage pre-built workflow actions ranging from provisioning site collections, list actions, interactions with external systems such as Exchange, Lync and SQL Server directly from your workflow .
- The ability to build collaboration workflows for document approvals and other line of business applications within your SharePoint farm.
- Powerful  web based drag- and- drop editing, allowing workflows to be developed in minimal time and effort.
- Powerful debugging capabilities.
- Integration with other Nintex products such as Nintex Live and Nintex Forms.

# An Introduction to Demand Management in Project Server 2010

## What is Demand Management?

With the release of Project Server 2010, a new capability called Demand Management was introduced. Borrowing from previous capabilities of Project Portfolio Server 2007, the Demand Management feature enables users to submit project proposals or ideas and then guide those ideas through a governance process. A key component in driving the governance process is the use of workflow; enabling data validation, approvals and implementation of business logic throughout the process.

Contrary to the name, the Demand Management capabilities of Project Server are not limited to determining the demand for resources and assessing project ideas. The framework the Demand Management capabilities put in place is available throughout the whole of the project lifecycle and is not limited to the initial stages of the project.

## Phases

Project Server uses the concept of phases to represent the high level stages of a project lifecycle, as can be seen in the figure below.

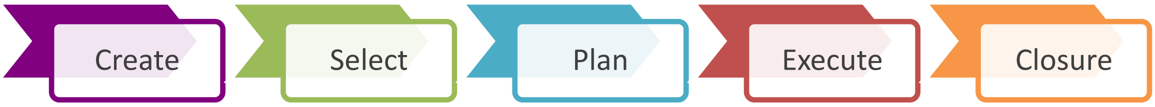


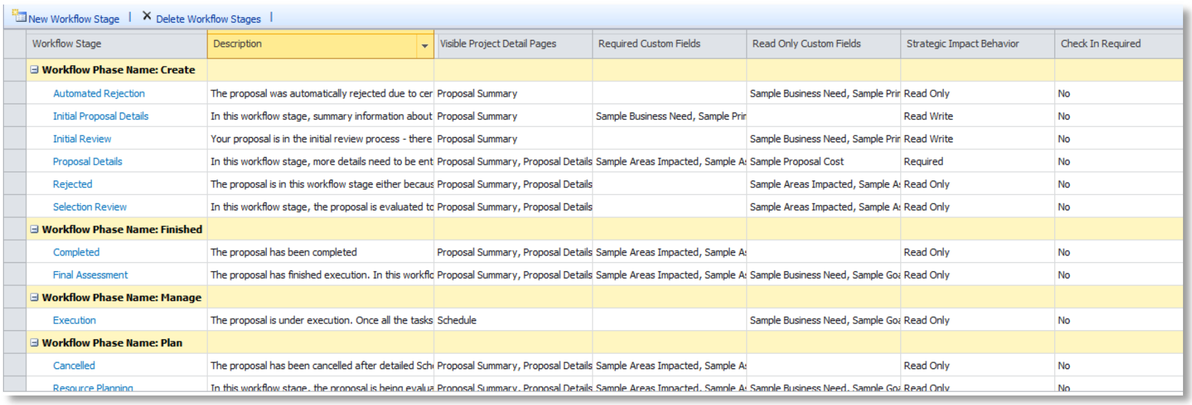
Figure 3: High Level Phases of a Project Lifecycle

Phases are used for grouping purposes within Project Server and can have many stages associated with it.

## Stages

A stage can be related to a single step within the project lifecycle and is the fundamental building block of the Demand Management capabilities in Project Server. Each stage has a number of attributes that can be configured which will determine how Project Server behaves. These attributes include:

- Which Project Detail Pages (PDP) should be visible
- What project custom fields must be populated within the stage
- What project custom fields should be read only
- Whether strategic impact values should be required, editable or read only
- Additional description and check in options



| Workflow Stage | Description | Visible Project Detail Pages | Required Custom Fields | Read Only Custom Fields | Strategic Impact Behavior | Check In Required |
|---|---|---|---|---|---|---|
| **Workflow Phase Name: Create** | | | | | | |
| Automated Rejection | The proposal was automatically rejected due to cer | Proposal Summary | | Sample Business Need, Sample Prir | Read Only | No |
| Initial Proposal Details | In this workflow stage, summary information about | Proposal Summary | Sample Business Need, Sample Prir | | Read Write | No |
| Initial Review | Your proposal is in the initial review process - there | Proposal Summary | | Sample Business Need, Sample Prir | Read Write | No |
| Proposal Details | In this workflow stage, more details need to be ent | Proposal Summary, Proposal Details | Sample Areas Impacted, Sample A: | Sample Proposal Cost | Required | No |
| Rejected | The proposal is in this workflow stage either becaus | Proposal Summary, Proposal Details | | Sample Areas Impacted, Sample A: | Read Only | No |
| Selection Review | In this workflow stage, the proposal is evaluated tc | Proposal Summary, Proposal Details | | Sample Areas Impacted, Sample A: | Read Only | No |
| **Workflow Phase Name: Finished** | | | | | | |
| Completed | The proposal has been completed | Proposal Summary, Proposal Details | Sample Areas Impacted, Sample A: | | Read Only | No |
| Final Assessment | The proposal has finished execution. In this workflc | Proposal Summary, Proposal Details | Sample Areas Impacted, Sample A: | Sample Business Need, Sample Go: | Read Only | No |
| **Workflow Phase Name: Manage** | | | | | | |
| Execution | The proposal is under execution. Once all the tasks | Schedule | | Sample Business Need, Sample Go: | Read Only | No |
| **Workflow Phase Name: Plan** | | | | | | |
| Cancelled | The proposal has been cancelled after detailed Sch | Proposal Summary, Proposal Details | Sample Areas Impacted, Sample A: | | Read Only | No |
| Resource Planning | In this workflow stage, the proposal is being evalua | Proposal Summary, Proposal Details | Sample Areas Impacted, Sample A: | Sample Business Need, Sample Go: | Read Only | No |

Figure 4 : Configuration of Project Server Demand Management Stages

## Project Details Pages

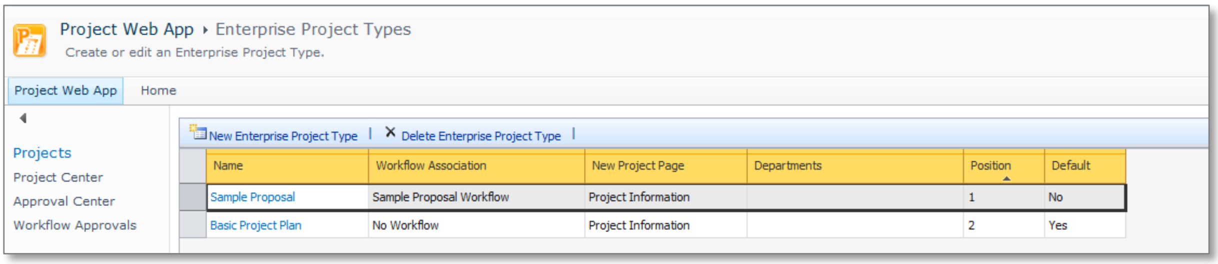
Project Detail Pages are SharePoint web part pages which can be configured to display and collect project related information through a number of custom web parts.

## Workflow

The key to the Demand Management functionality is a workflow that moves the governance process from stage to stage, ensuring business rules and validations are followed so the workflow meets the various business rules of the organization. Project Server workflows leverage a number of special actions that enable the workflow to view and update data associated with the project, such as project custom fields and reading project server specific security groups. When merged with standard SharePoint workflow functionality, it is possible to build more complex workflows incorporating items such as approvals, branching logic and integration with external services.

## Enterprise Project Types

Once built, each workflow can be assigned to a specific type of project, allowing different workflows to be used for different project types within Project Server. For instance some organizations have different project governance processes depending on the identified risk of the project. Through Enterprise Project Types, it is possible to create a type of project which is associated with its own unique workflow.



Project Web App ▸ Enterprise Project Types
Create or edit an Enterprise Project Type.
Project Web App
Home
Projects
Project Center
Approval Center
Workflow Approvals
New Enterprise Project Type
Delete Enterprise Project Type

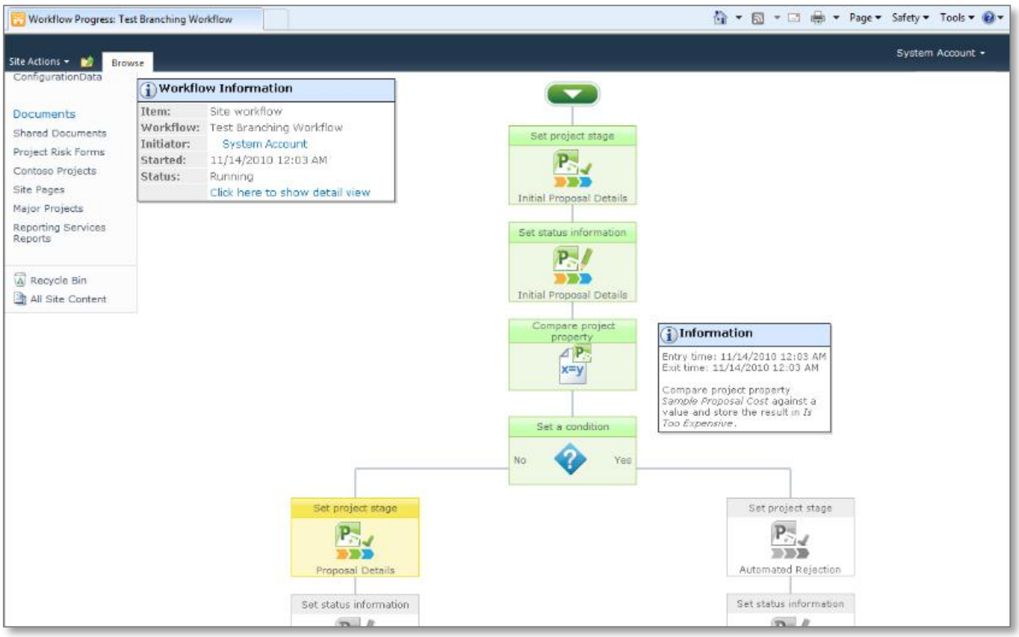
| Name | Workflow Association | New Project Page | Departments | Position | Default |
|---|---|---|---|---|---|
| Sample Proposal | Sample Proposal Workflow | Project Information | | 1 | No |
| Basic Project Plan | No Workflow | Project Information | | 2 | Yes |

Figure 5 : Enterprise Project Type

# Nintex Workflow for Project Server

Initially, building Project Server workflows required the use of Microsoft Visual Studio 2010. Developers were provided with a number of new workflow actions which could be used as the building blocks for building Project Server Demand Management workflows. These actions provided capabilities such as reading and updating Project custom fields, reading Project Server security groups, changing the project stage as well as numerous other Project Server workflow events.

Developing workflows in Visual Studio 2010 whilst powerful are very complex to build, requiring organizations to have skilled resources with staff who have both technical knowledge of Project Server as well as building Visual Studio workflows. The development of these workflows can also take a lot of time to design, build and test.  Nintex recognized this limitation and set out to simplify the development of Project Server workflows by developing a simple web based drag-and-drop tool for building workflows that can be used to empower an organization with  minimum overhead.



Workflow Information

Item: Site workflow
Workflow: Test Branching Workflow
Initiator: System Account
Started: 11/14/2010 12:03 AM
Status: Running
Click here to show detail view

Information

Entry time: 11/14/2010 12:03 AM
Exit time: 11/14/2010 12:03 AM

Compare project property *Sample Proposal Cost* against a value and store the result in *Is Too Expensive*.

Figure 6 : Nintex Workflow for Project Server 2010

In 2010, Nintex released the first version of Nintex Workflow for Project Server, a landmark product that empowered organizations to build Demand Management Workflows quickly and easily for their Project Server 2010 investment.

In 2012, Nintex released Version 2.0 of the product containing a number of new capabilities further unlocking the potential of Project Server within the Enterprise.

# Building Demand Management Workflows with Nintex Workflow for Project Server

## Planning Your Workflow

Before commencing development of a Demand Management Workflow with Nintex, it is important to take time to plan what is required. Demand Management Workflows exist in partnership with the configuration of Project Server, meaning that it is necessary to have determined the phases, stages, project detail pages and custom field requirements for the workflow. Only once these have been determined and configured within Project Server is it possible to build the workflow to thread these items together.

This partnership also introduces additional complexity when deploying workflows, as it is necessary to ensure the same project server configuration is available in the target environment as used in the development environment. Microsoft has a number of tools to assist with the deployment of these Demand Management assets, such as Playbooks and the Demand Management Export & Import Tool.

Please refer to the references section of this document for links to these tools.

## Creating a Simple Approval Demand Management Workflow

For the purpose of this guide, we will demonstrate how to build two Demand Management Workflows: simple approval workflow and advanced approval workflow.

**Simple Approval Workflow**

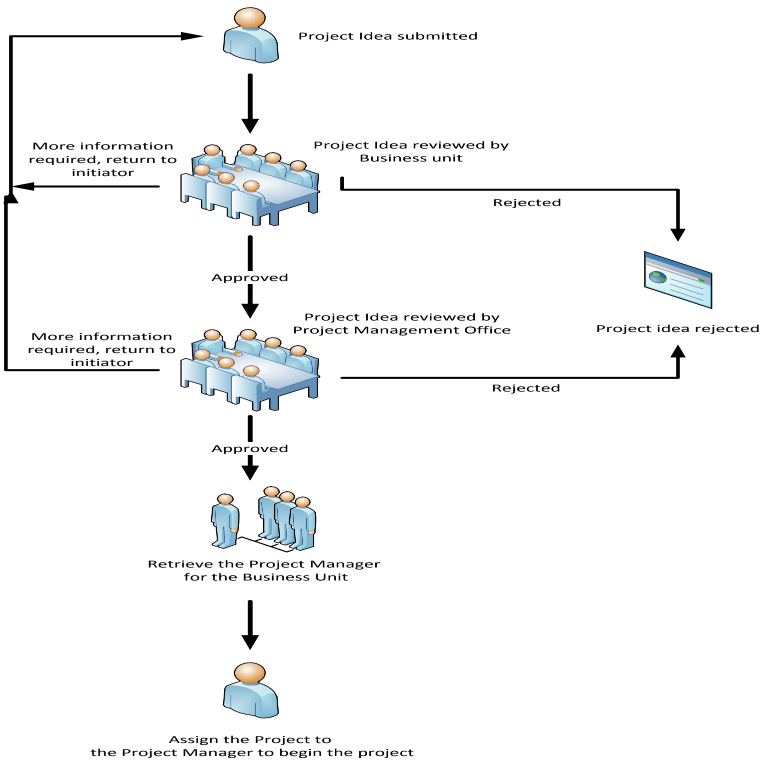

Project Idea submitted
More information required, return to initiator
Project Idea reviewed by Business unit
Rejected
Approved
Project Idea reviewed by Project Management Office
Project idea rejected
More information required, return to initiator
Rejected
Approved
Retrieve the Project Manager for the Business Unit
Assign the Project to the Project Manager to begin the project

Figure 7: Simple Demand Management Workflow

The workflow will begin with project proposal, determine the members of the Project Server Portfolio Management Group, raise an approval task for each member of the group and depending on the outcome, either approve or reject the project proposal.

**Note:** Prior to building this workflow, confirm that the Portfolio Managers Group within Project Server has at least one user assigned to it.

## Building the Workflow

1. To create a Demand Management Workflow, click on **Site Actions** > **Project Server Workflows** > **Create Demand Management Workflow**.

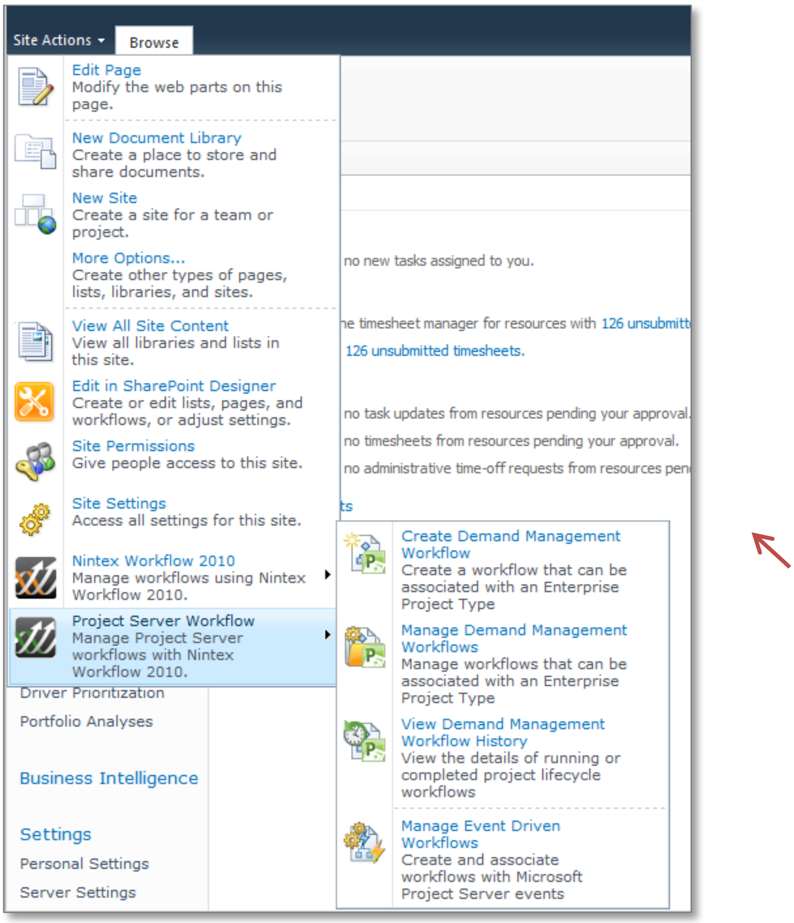
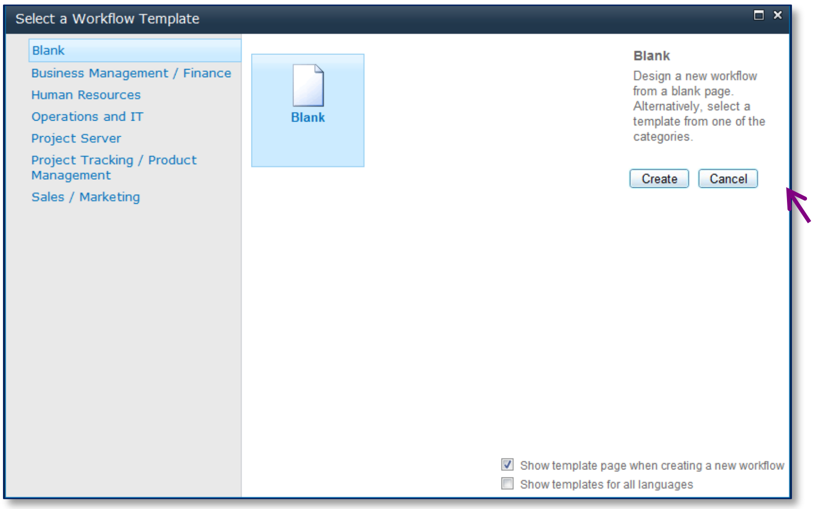

Site Actions
Browse
Edit Page
Modify the web parts on this page.
New Document Library
Create a place to store and share documents.
New Site
Create a site for a team or project.
More Options...
Create other types of pages, lists, libraries, and sites.
View All Site Content
View all libraries and lists in this site.
Edit in SharePoint Designer
Create or edit lists, pages, and workflows, or adjust settings.
Site Permissions
Give people access to this site.
Site Settings
Access all settings for this site.
Nintex Workflow 2010
Manage workflows using Nintex Workflow 2010.
Project Server Workflow
Manage Project Server workflows with Nintex Workflow 2010.
Driver Prioritization
Portfolio Analyses
Business Intelligence
Settings
Personal Settings
Server Settings
no new tasks assigned to you.
he timesheet manager for resources with 126 unsubmitt
126 unsubmitted timesheets.
no task updates from resources pending your approval.
no timesheets from resources pending your approval.
no administrative time-off requests from resources pen
ts
Create Demand Management Workflow
Create a workflow that can be associated with an Enterprise Project Type
Manage Demand Management Workflows
Manage workflows that can be associated with an Enterprise Project Type
View Demand Management Workflow History
View the details of running or completed project lifecycle workflows
Manage Event Driven Workflows
Create and associate workflows with Microsoft Project Server events

Figure 8: Demand Management Workflow Menu

2. In the Workflow Template dialog, select **Blank** and click on **Create**.


Select a Workflow Template
Blank
Business Management / Finance
Human Resources
Operations and IT
Project Server
Project Tracking / Product Management
Sales / Marketing
Blank
Create
Cancel

Blank

Design a new workflow from a blank page. Alternatively, select a template from one of the categories.

☑ Show template page when creating a new workflow

□ Show templates for all languages

Figure 9: Select Workflow Template

Nintex Workflow for Project Server provides thirteen workflow actions specifically used to build Demand Management Workflows.

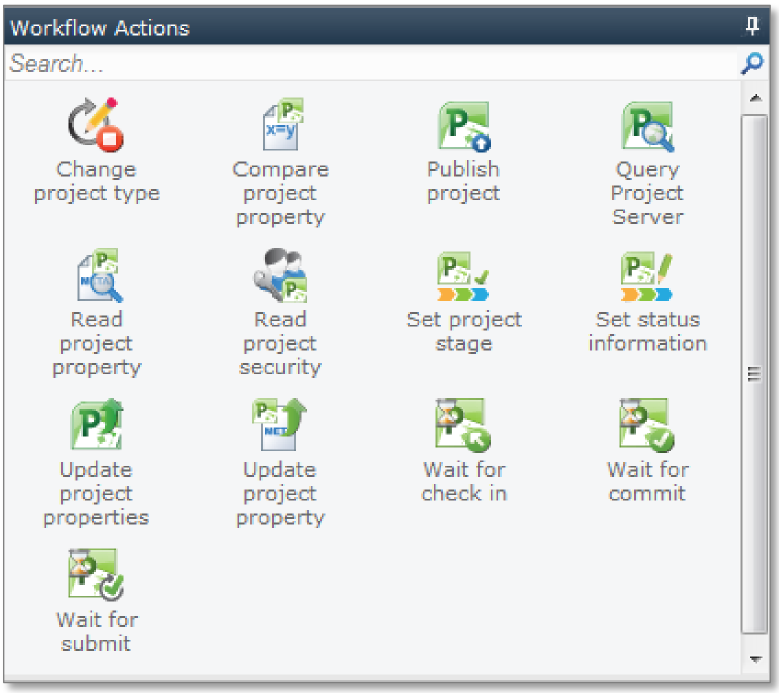


Figure 10: Project Server Workflow Actions

The first action we will use is the **Set project stage** action. Project Server will read the associated configuration and ensure the correct Project Detail Pages, Custom Field Validation; Read Only Custom fields etc. are enabled in the Project Server web pages.

3. Select and drag a **Set project stage** workflow action onto the workflow canvas.

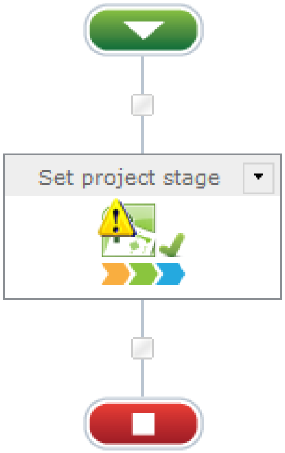
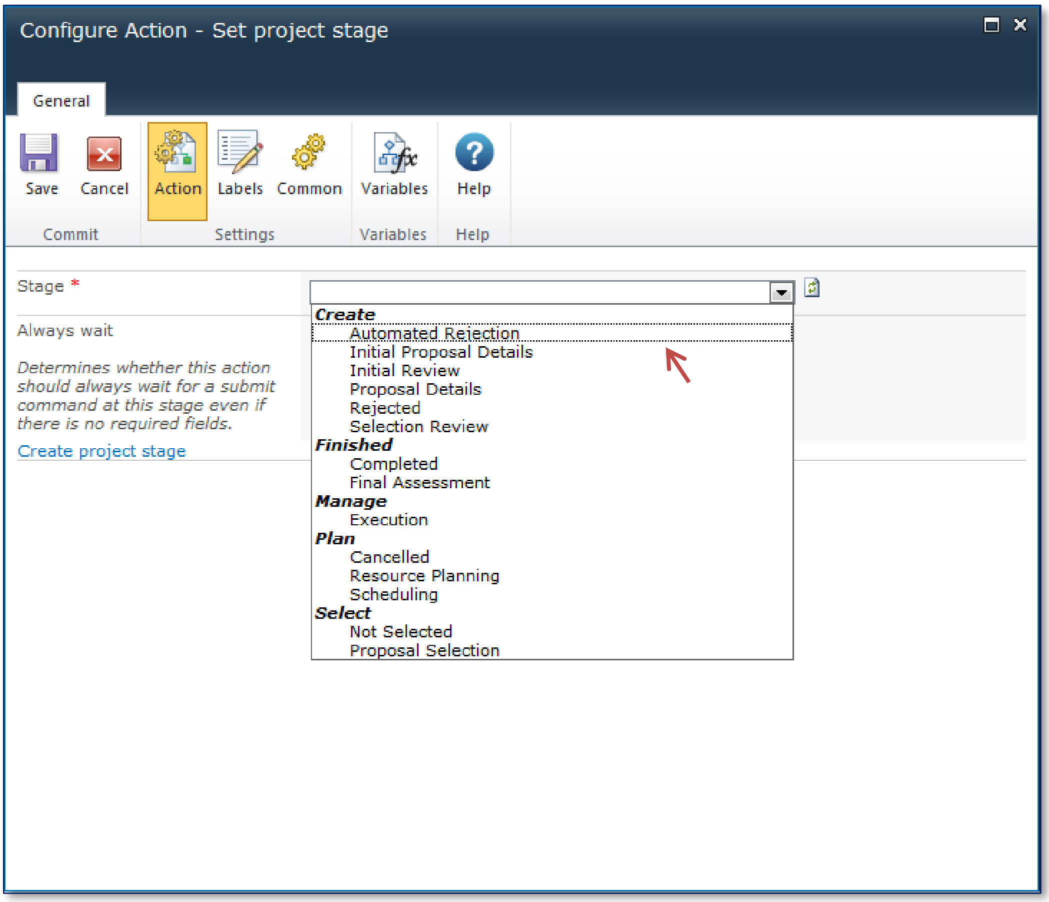


Figure 11: Set Project Stage

4. Double-click on the **Set project stage** action to open the configuration dialog. The dialog will show all the stages configured within the PWA instance.



Stage *

Always wait

*Determines whether this action should always wait for a submit command at this stage even if there is no required fields.*

Create project stage

***Create***
- Automated Rejection
- Initial Proposal Details
- Initial Review
- Proposal Details
- Rejected
- Selection Review

***Finished***
- Completed
- Final Assessment

***Manage***
- Execution

***Plan***
- Cancelled
- Resource Planning
- Scheduling

***Select***
- Not Selected
- Proposal Selection

Figure 12: Configure Project Stage

5. In the **Stage** field, select **Initial Proposal Details** and click **Save**.

The next step is to select a **Read project security group** workflow action. This will read the members of the Project Server Portfolio Managers security group. By default this group will not have any members, so you will need to manually add users to the group via **Server Settings** > **Manage Groups**.

6. Select a **Read project security group** workflow action and place on the workflow canvas.

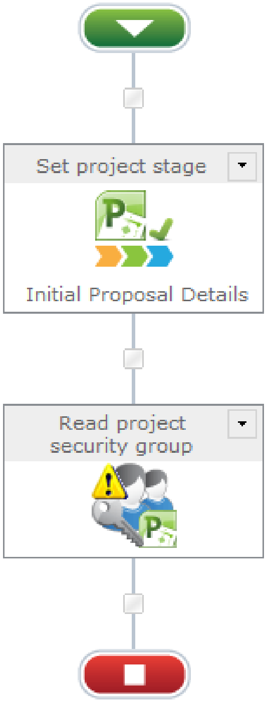
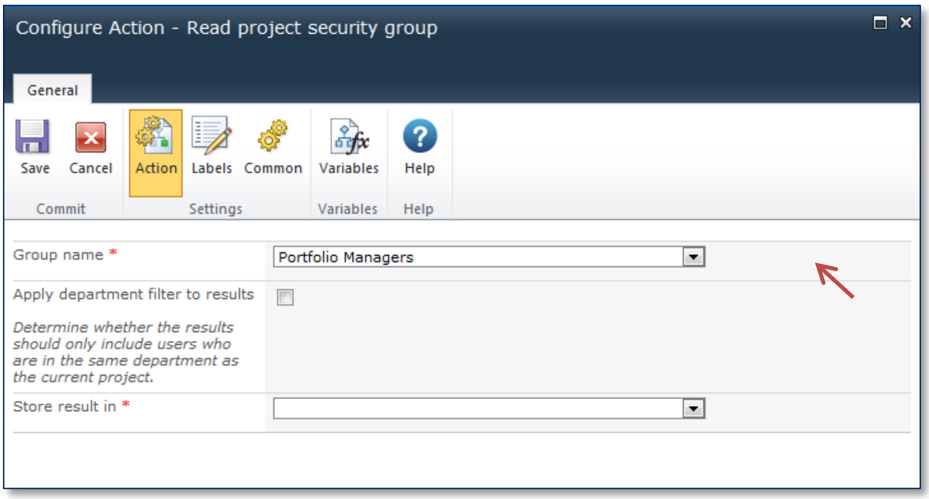


Figure 7 : Read Project Security Group

7. Double-click on the **Read project security group** action to configure.
   a. In the **Group Name** field, select **Portfolio Managers** from the menu.



Group name * Portfolio Managers

Apply department filter to results □

*Determine whether the results should only include users who are in the same department as the current project.*

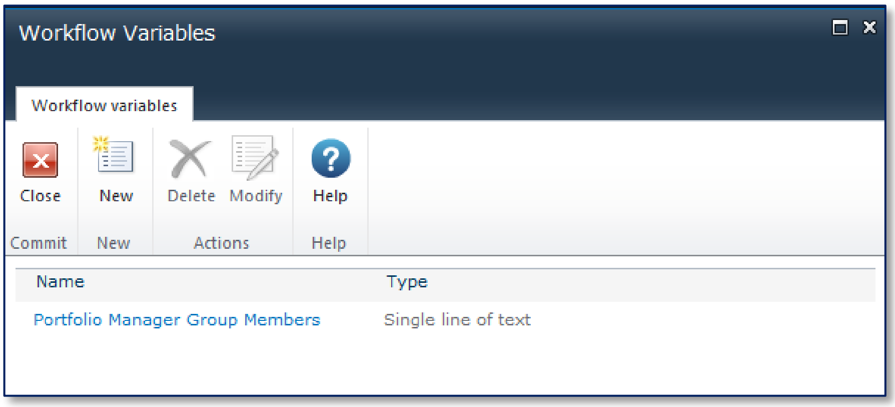
Store result in *

Figure 14: Read Project Security Group Configuration

   b. For the workflow to use the data, it is necessary to store the result in a variable.

To create a variable:

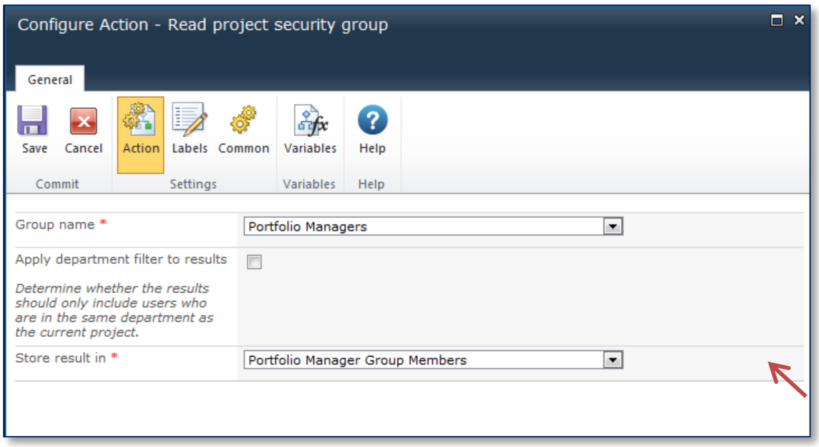
i. In the Ribbon, select **Variables.**
ii. Click **New.**
iii. Create a **Single Line of Text** variable called **Portfolio Managers Group Members**.
iv. Click **Save** and **Close.**



| Name | Type |
|---|---|
| Portfolio Manager Group Members | Single line of text |

Figure 8 : Create Workflow Variable

c. In the **Store Result in** field, select the **Portfolio Managers Group Members** variable.



Group name * Portfolio Managers

Apply department filter to results □

*Determine whether the results should only include users who are in the same department as the current project.*

Store result in * Portfolio Manager Group Members

Figure 16: Configured Read Project Security Group

d. Click **Save**.

The next action will enable us to create approval tasks for each member of the Portfolio Managers security group and depending on the outcome of those approvals, route the workflow to either approve the proposal, or reject it.

8. To do this, select an **Assign Flexi Task** action and place on the workflow canvas below the **Read project security group** action.

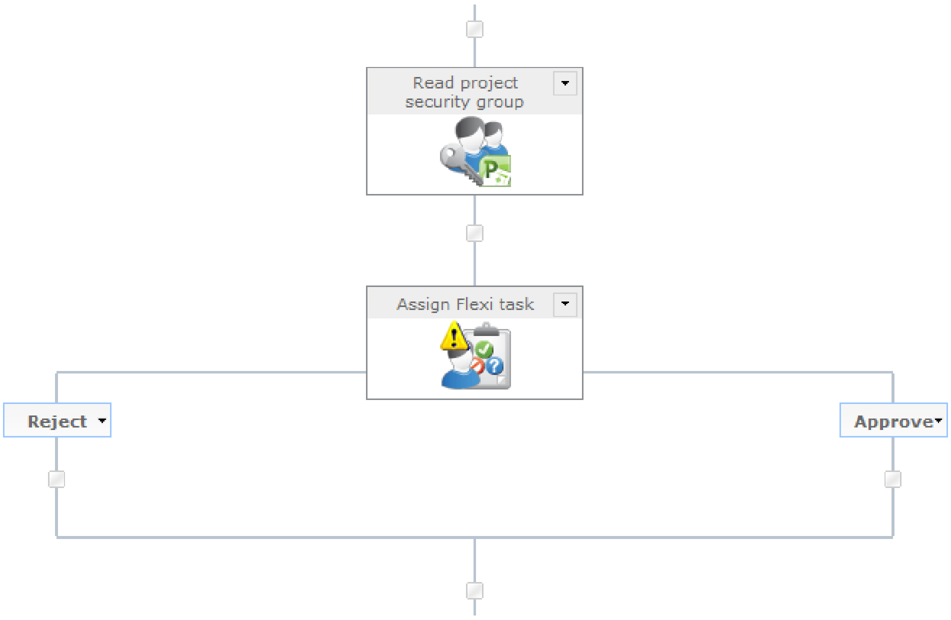
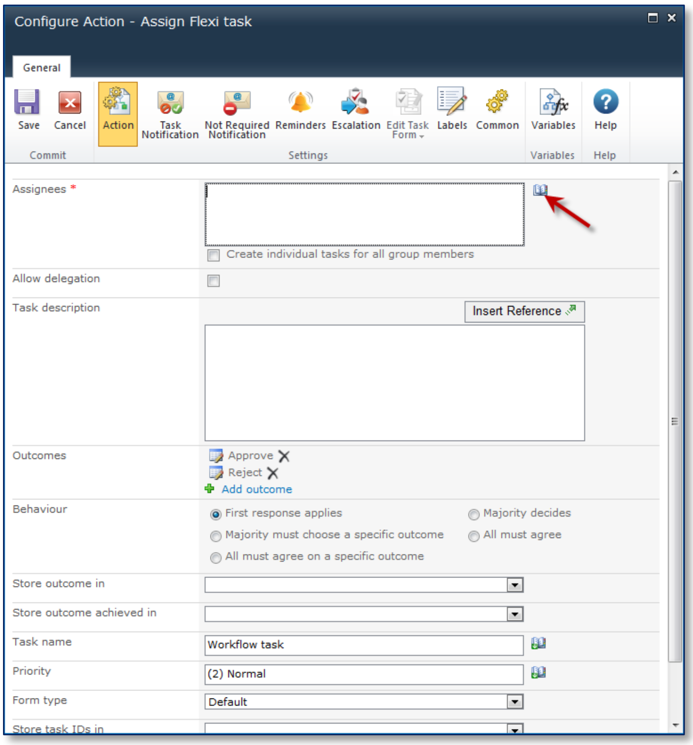


Figure 97: Flexi Task in Workflow

By default, the **Assign Flexi Task** action will provide two branches, Reject and Approve, but the action can be configured to provide additional branches if required.

9. Double-click on the **Assign Flexi task** to open the configuration dialog.
    a. In the **Assignees** field, click the **Lookup Icon**.



Assignees *

□ Create individual tasks for all group members

Allow delegation □

Task description

Outcomes: Approve, Reject

Behaviour:
● First response applies
○ Majority decides
○ Majority must choose a specific outcome
○ All must agree
○ All must agree on a specific outcome

Store outcome in

Store outcome achieved in

Task name: Workflow task

Priority: (2) Normal

Form type: Default

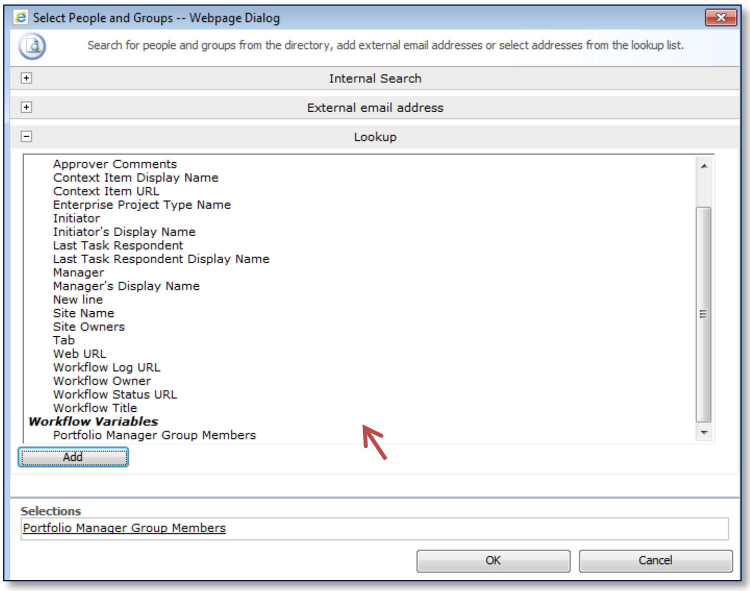
Store task IDs in

Figure 108: Flexi Task Assignees Lookup

Back to Contents

b. Select the **Lookup** section and scroll to the bottom. Click on **Portfolio Manager Group Members**

c. Click **Add** and then **OK**.



Select People and Groups -- Webpage Dialog

Search for people and groups from the directory, add external email addresses or select addresses from the lookup list.

Approver Comments
Context Item Display Name
Context Item URL
Enterprise Project Type Name
Initiator
Initiator's Display Name
Last Task Respondent
Last Task Respondent Display Name
Manager
Manager's Display Name
New line
Site Name
Site Owners
Tab
Web URL
Workflow Log URL
Workflow Owner
Workflow Status URL
Workflow Title
***Workflow Variables***
Portfolio Manager Group Members

Selections
Portfolio Manager Group Members

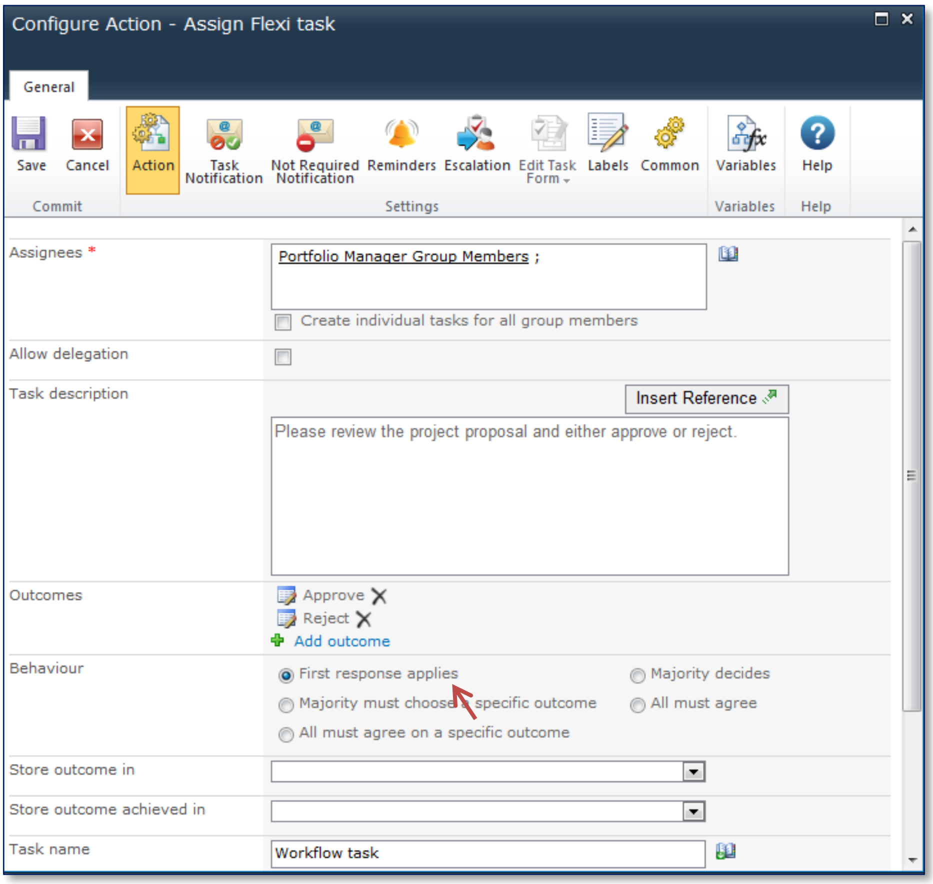
**Figure 19: Select People and Groups**

The **Assign Flexi task** action contains a number of options which makes it much more powerful than a typical office task, such as configuring reminder alerts and escalations if required. For this example, we will keep the workflow simple and will not configure these options.

10. In the **Task Description** field type, “Please review the project proposal and either approve or reject”. This description will be included in any tasks the **Assign Flexi task** action creates.

You can also configure the behavior of how the task will behave when it receives outcomes from the various parties it is assigned to.

11. In the **Behavior** field, select **First response applies** and then click **Save**.



Assignees * Portfolio Manager Group Members ;

□ Create individual tasks for all group members

Allow delegation □

Task description

Insert Reference

Please review the project proposal and either approve or reject.

Outcomes

Approve

Reject

Add outcome

Behaviour

● First response applies ○ Majority decides

○ Majority must choose a specific outcome ○ All must agree

○ All must agree on a specific outcome

Store outcome in

Store outcome achieved in

Task name Workflow task

Figure 20: Configured Flexi Task

The next step is to configure the **Reject** branch of the **Assign Flexi task**. We will set the project stage to be **Rejected** and then terminate the workflow. Once completed, the rejected branch of the workflow should look like the image in **Figure 11 : Rejected Branch of Flexi** Task.

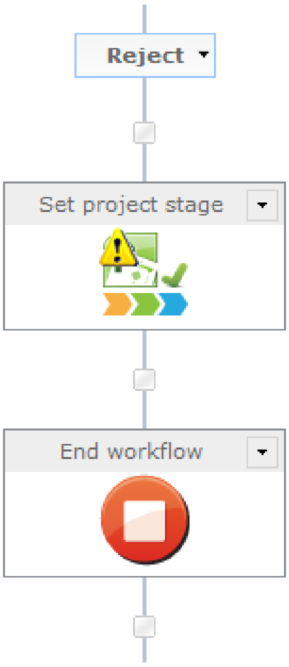
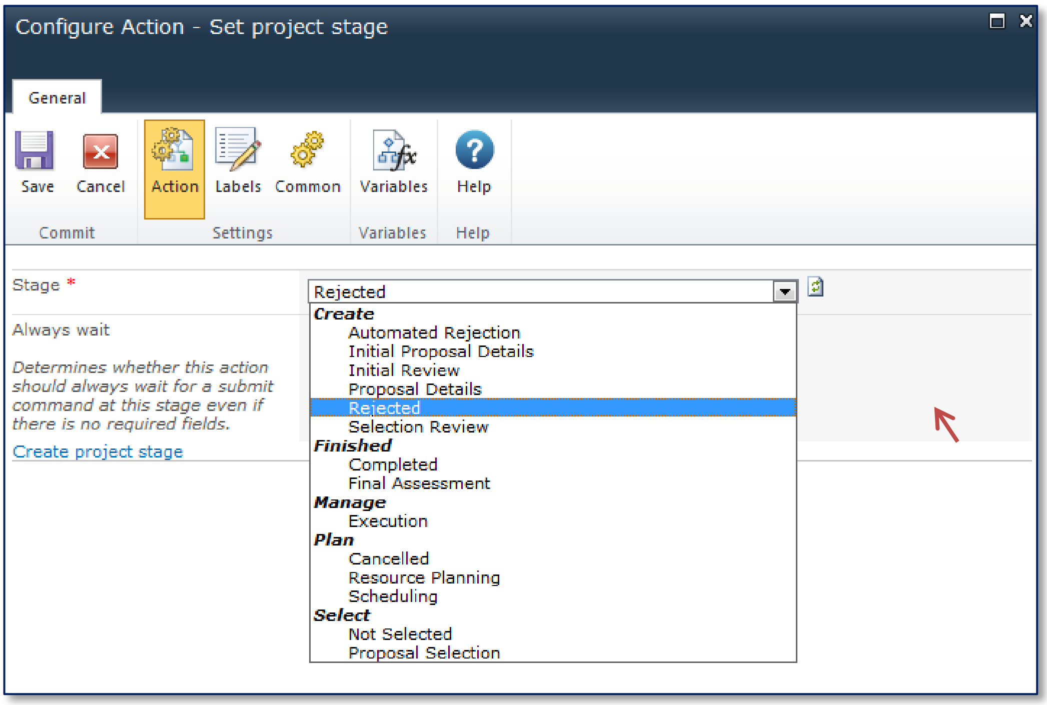


Figure 11 : Rejected Branch of Flexi Task

12. Double-click on the **Set project stage** action to open the configuration screen.
    a. In the **Project Stage** field, select **Rejected**.



Stage * Rejected

Always wait

*Determines whether this action should always wait for a submit command at this stage even if there is no required fields.*

Create project stage

**Create**
- Automated Rejection
- Initial Proposal Details
- Initial Review
- Proposal Details
- Rejected
- Selection Review

**Finished**
- Completed
- Final Assessment

**Manage**
- Execution

**Plan**
- Cancelled
- Resource Planning
- Scheduling

**Select**
- Not Selected
- Proposal Selection

Figure 22: Configure Rejected Stage

Configure the Approve branch of the **Assign Flexi Task** action.

13. Select a **Set Project Stage** workflow action and place on the approved branch of the **Assign Flexi Task** action.

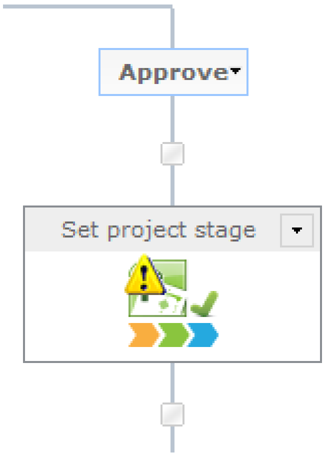
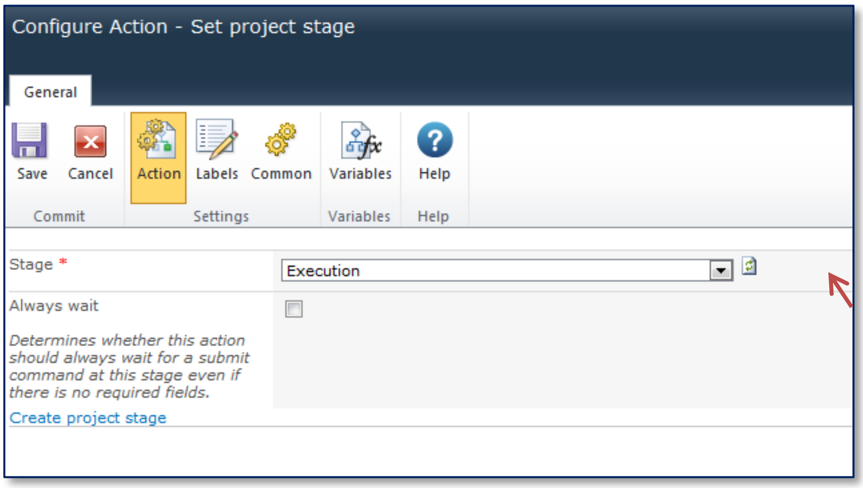


Figure 23: Approve Branch of Flexi Task

14. Double-click on the action to open the configuration dialog.
    a. In the **Stage** field, select **Execution.**



Stage * Execution

Always wait □

*Determines whether this action should always wait for a submit command at this stage even if there is no required fields.*

Create project stage

Figure 24: Configure Execution Stage

Finally, **Publish** the workflow.

15. In the Nintex Workflow 2010 Ribbon, click **Publish.**

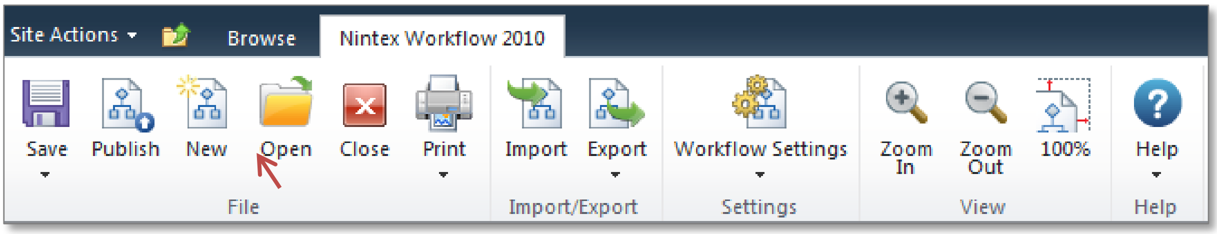
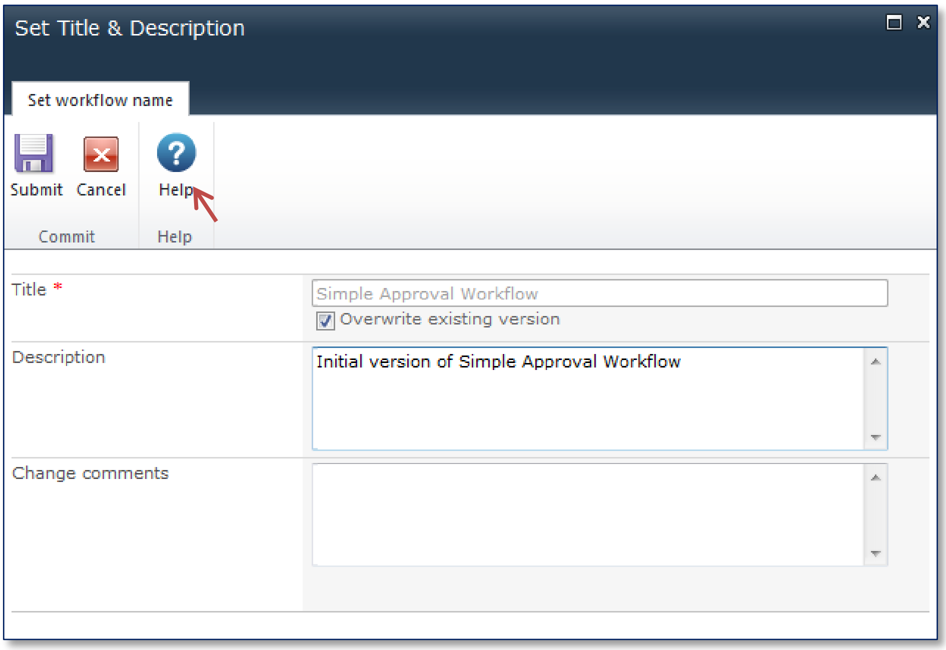


Figure 25: Nintex Workflow Ribbon

16. In the **Title** field type, 'Simple Approval Workflow'.
17. Click **Submit.**



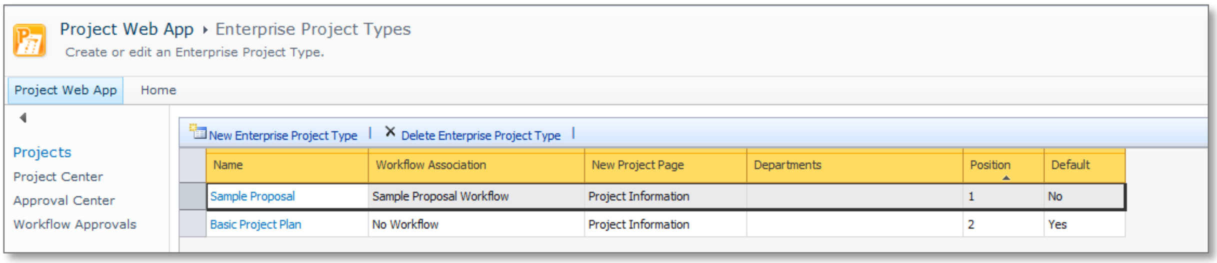
Title * Simple Approval Workflow
☑ Overwrite existing version
Description Initial version of Simple Approval Workflow
Change comments

Figure 26: Publish Workflow

Once the workflow has been published, it will be available for selection in the Enterprise Project Type association screen.

## Deploying a Demand Management Workflow

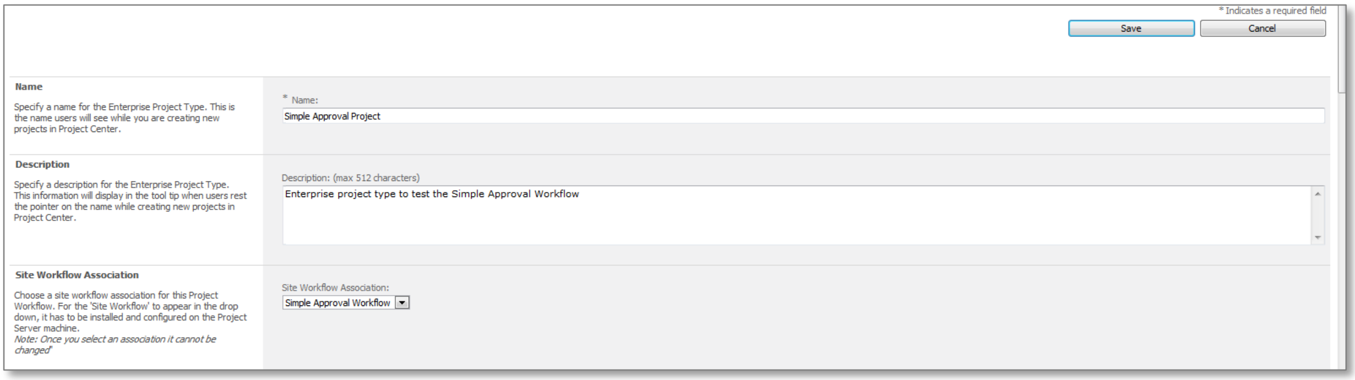
Before you can use a Demand Management Workflow it needs to be associated with an Enterprise Project Type (EPT). The EPT outlines the various components of the project type including workflow, schedule template and project workspace template.


Project Web App ▸ Enterprise Project Types
Create or edit an Enterprise Project Type.
Project Web App
Home
Projects
Project Center
Approval Center
Workflow Approvals
New Enterprise Project Type
Delete Enterprise Project Type

| Name | Workflow Association | New Project Page | Departments | Position | Default |
|---|---|---|---|---|---|
| Sample Proposal | Sample Proposal Workflow | Project Information | | 1 | No |
| Basic Project Plan | No Workflow | Project Information | | 2 | Yes |

Figure 27: Enterprise Project Types

1. To create a new EPT, navigate to **Server Settings** > **Enterprise Project Types** and select **New Enterprise Project Type.**
2. In the **Name** field type, 'Simple Approval Project'.
3. In the **Site Workflow Association** field, select **Simple Approval Workflow.**



**Name**

Specify a name for the Enterprise Project Type. This is the name users will see while you are creating new projects in Project Center.

* Name: Simple Approval Project

**Description**

Specify a description for the Enterprise Project Type. This information will display in the tool tip when users rest the pointer on the name while creating new projects in Project Center.

Description: (max 512 characters) Enterprise project type to test the Simple Approval Workflow

**Site Workflow Association**

Choose a site workflow association for this Project Workflow. For the 'Site Workflow' to appear in the drop down, it has to be installed and configured on the Project Server machine.
*Note: Once you select an association it cannot be changed"*

Site Workflow Association: Simple Approval Workflow

Figure 28: Create a New Enterprise Project Type

4. Click **Save** so the EPT is available for use.

5. To check if the EPT has been configured successfully, navigate to Project Centre and click on **New Project** in the Ribbon. The new EPT should be visible in the drop-down list.

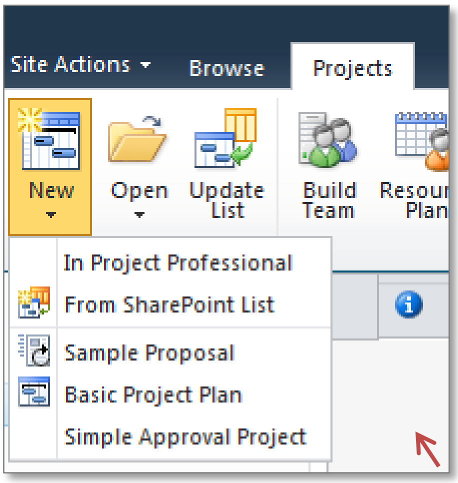
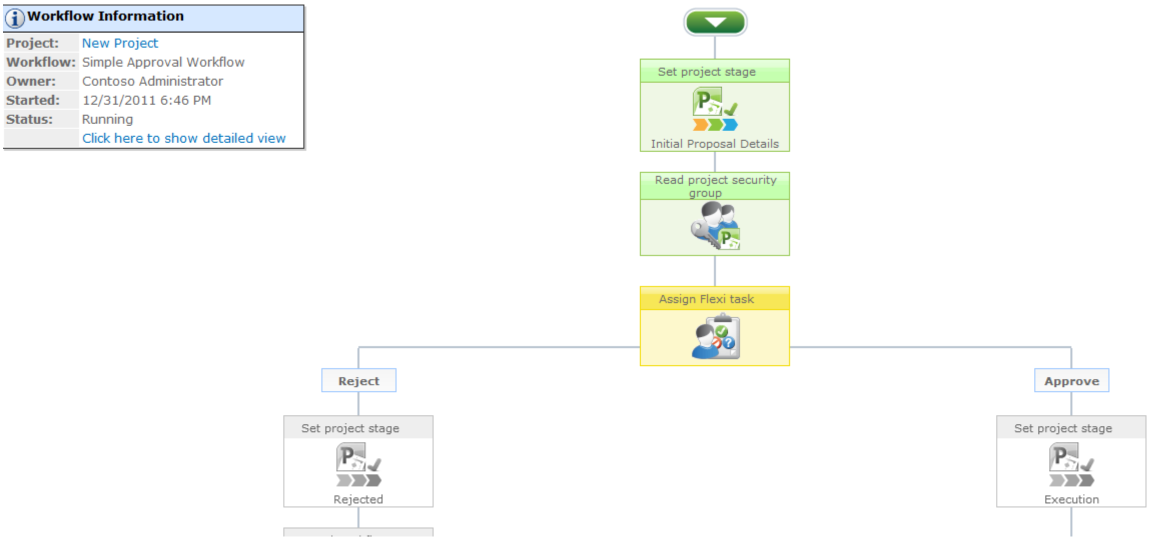


Figure 29: Project Centre New Menu

## Viewing the Status of a Workflow

Nintex Workflow for Project Server extends the normal Demand Management Workflow capabilities to provide a visual status representation. For convenience, a Workflow Status button has been added to the Ribbon. This button is visible when a user is editing a workflow controlled project. Click on the **Workflow Status** button to view a graphical representation of the workflow.



**Workflow Information**

| | |
|---|---|
| **Project:** | New Project |
| **Workflow:** | Simple Approval Workflow |
| **Owner:** | Contoso Administrator |
| **Started:** | 12/31/2011 6:46 PM |
| **Status:** | Running |
| | Click here to show detailed view |

Figure 30: Workflow Status

The workflow actions that have been completed are green, the current stage of the workflow is yellow and those yet to be processed are grey. In the example above, the workflow has completed the first two stages and is waiting for the tasks to be approved or rejected.

## Creating an Advanced Demand Management Workflow

To see more of the benefits of using Nintex Workflow for Project Server 2010 to build your Demand Management Workflow, we need to create a more advanced workflow. For this example, we will build on the concepts in the previous example and create a more advanced workflow as outlined in the diagram below.

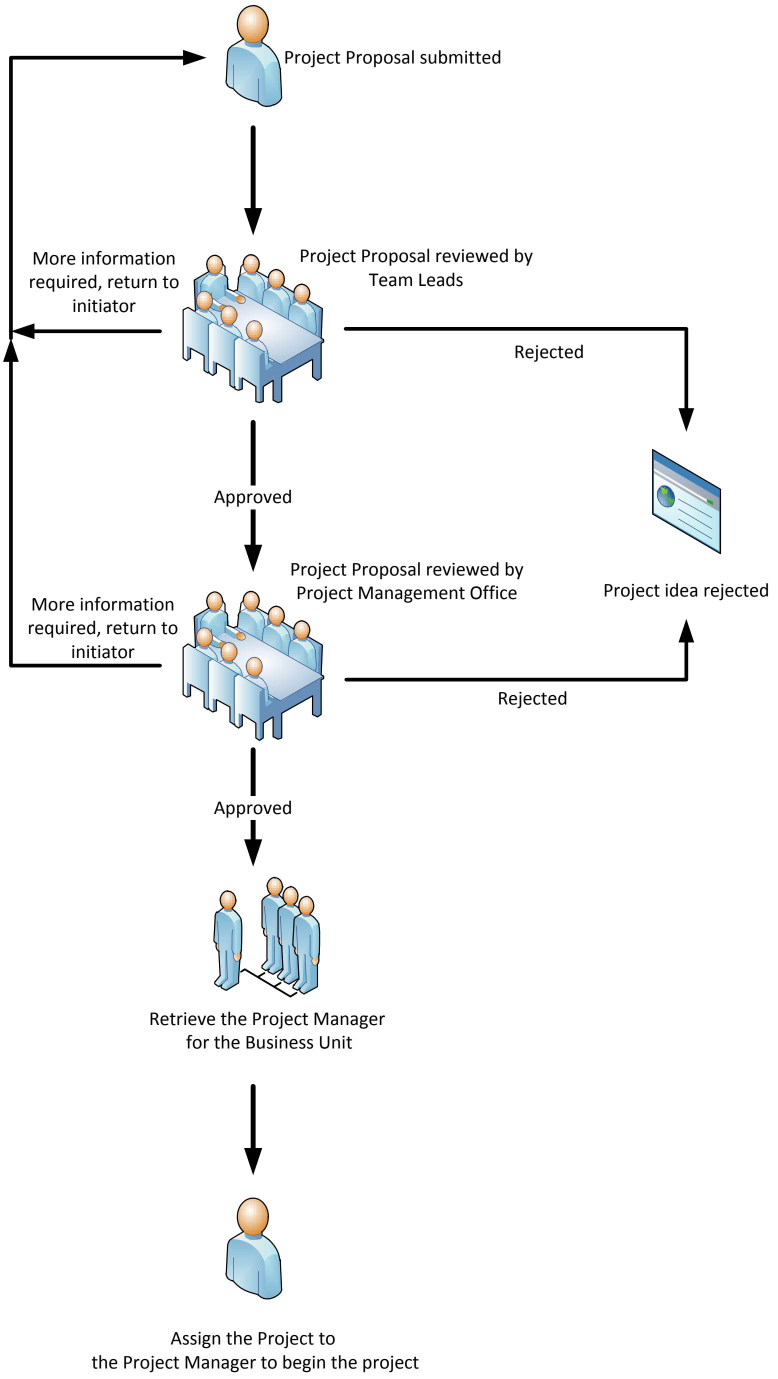

Project Proposal submitted
More information required, return to initiator
Project Proposal reviewed by Team Leads
Rejected
Approved
Project Proposal reviewed by Project Management Office
More information required, return to initiator
Project idea rejected
Rejected
Approved
Retrieve the Project Manager for the Business Unit
Assign the Project to the Project Manager to begin the project

Figure 31: Complex Demand Management Workflow

As you can see above, the workflow has two levels of approval, each with three possible outcomes:

- Approve
- Reject or
- Request more information

Should more information be required at any stage, the workflow will notify the proposer to provide more information and begin the approval process again. If the proposal is approved, then the workflow will perform a lookup in Project Server to find the relevant Project Manager for the business unit and change the project owner to that Project Manager.

## Project Server Configuration

Before building this example, it is necessary to configure a few additional options within Project Server. The configuration used in this example is based on the base configuration automatically provisioned in a new Project Server instance.

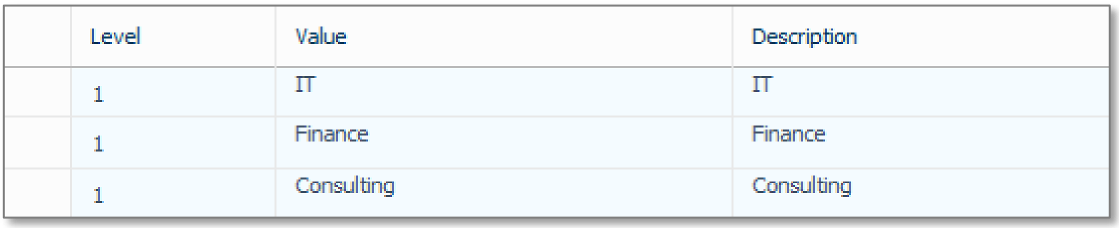
## Manage Groups

Please verify the following Project Server groups have at least one user associated per group:

- Team Leads
- Portfolio Managers

## Lookup Tables

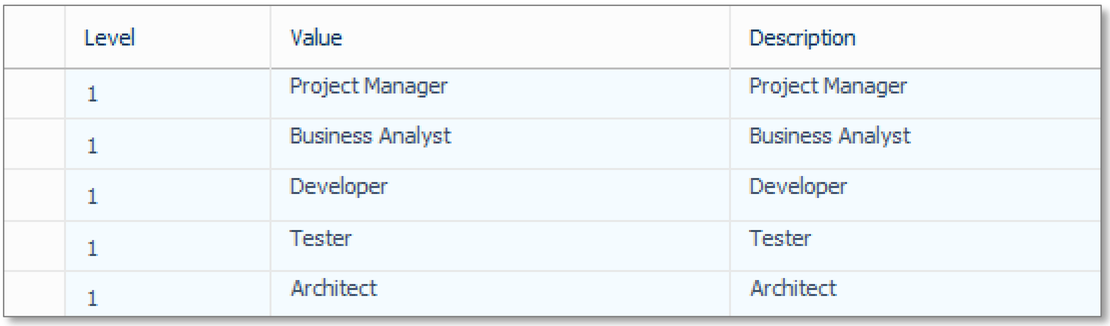
1. Create a new lookup table called **Business unit** and enter the following values:



| Level | Value | Description |
|---|---|---|
| 1 | IT | IT |
| 1 | Finance | Finance |
| 1 | Consulting | Consulting |

Figure 32: Business Unit Lookup Table

2. Create a second lookup table called **Position Role** and enter the following values.



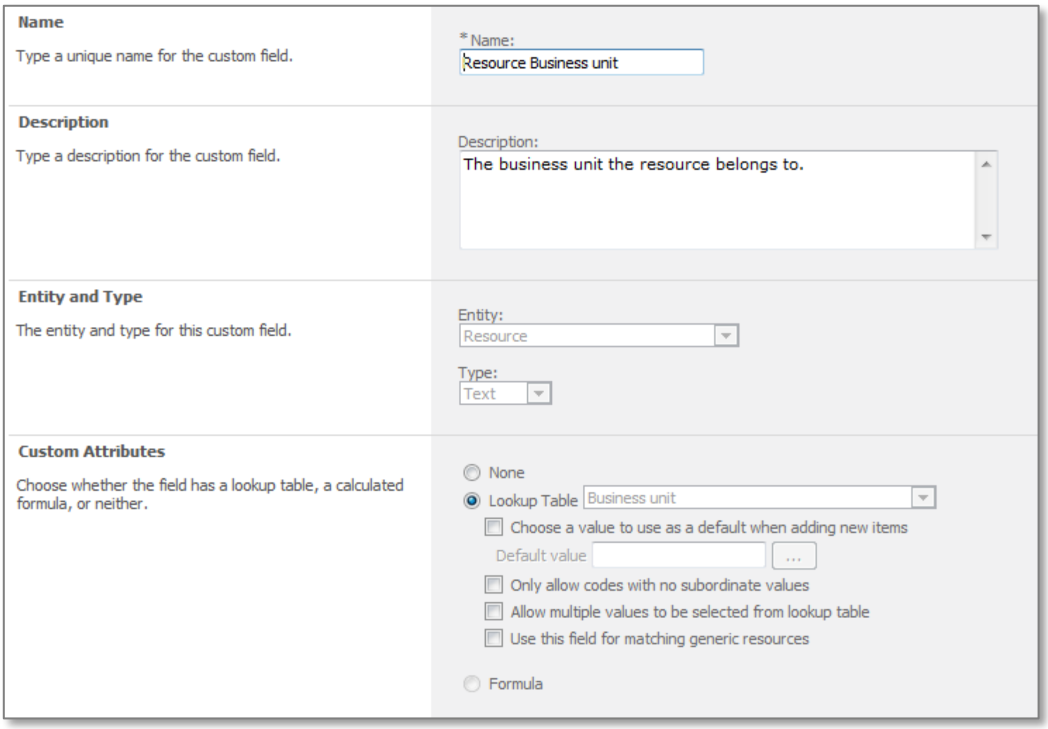
| Level | Value | Description |
|---|---|---|
| 1 | Project Manager | Project Manager |
| 1 | Business Analyst | Business Analyst |
| 1 | Developer | Developer |
| 1 | Tester | Tester |
| 1 | Architect | Architect |

Figure 33: Position Role Lookup Table

## Custom Fields

Create the following additional custom fields:

a. Create a Resource Custom Field called **Resource Business unit** and configure it as outlined in the screenshot below. This field will be used to assign business units to our resources.
   a. In the **Name** field type, "Resource Business unit".
   b. In the **Description** field type, "The business unit the resource belongs to".
   c. In the **Entity and Type** field, select the **Resource** entity and type **Text**.
   d. In the **Custom Attributes** section, select **Lookup Table** and choose the **Business unit** lookup table.



**Name**

Type a unique name for the custom field.

* Name: Resource Business unit

**Description**

Type a description for the custom field.

Description: The business unit the resource belongs to.

**Entity and Type**

The entity and type for this custom field.

Entity: Resource

Type: Text

**Custom Attributes**

Choose whether the field has a lookup table, a calculated formula, or neither.

○ None
● Lookup Table Business unit
□ Choose a value to use as a default when adding new items
Default value
□ Only allow codes with no subordinate values
□ Allow multiple values to be selected from lookup table
□ Use this field for matching generic resources
○ Formula

Figure 34: Resource Business Unit Custom Field

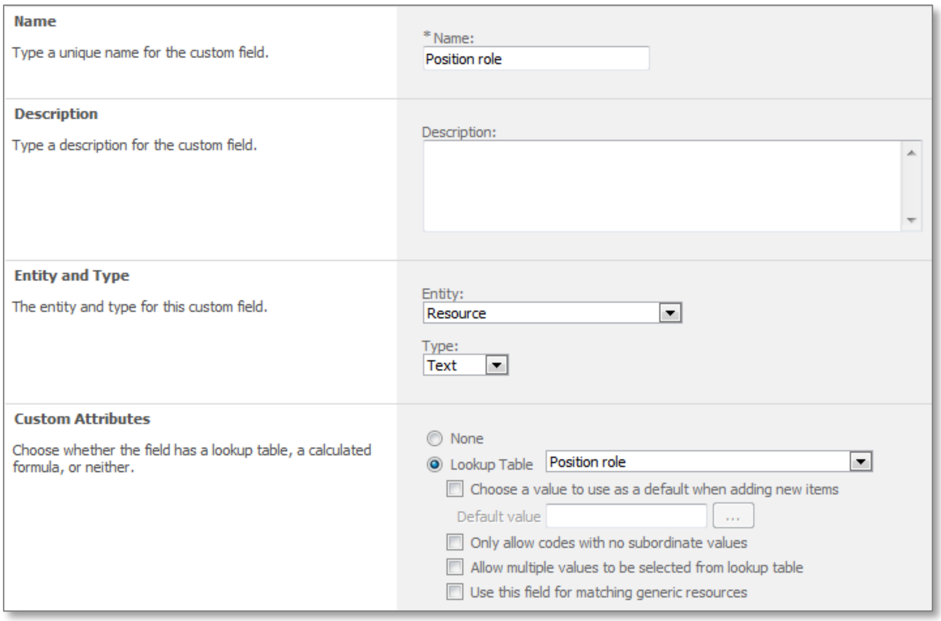
b. Create another resource custom field called **Position role**.

   a. In the **Name** field type, "Position role". This will store the position of the resource.

   b. In the **Custom Attributes** field, click **Lookup Table** and select **Position Role.**

   c. In the **Entity and Type** field, select the **Resource** entity and type **Text**.

   d. In the **Custom Attributes** section, select **Lookup Table** and choose the **Position role** lookup table.



**Name**

Type a unique name for the custom field.

* Name: Position role

**Description**

Type a description for the custom field.

Description:

**Entity and Type**

The entity and type for this custom field.

Entity: Resource

Type: Text

**Custom Attributes**

Choose whether the field has a lookup table, a calculated formula, or neither.

○ None

● Lookup Table Position role

□ Choose a value to use as a default when adding new items

Default value ...

□ Only allow codes with no subordinate values

□ Allow multiple values to be selected from lookup table

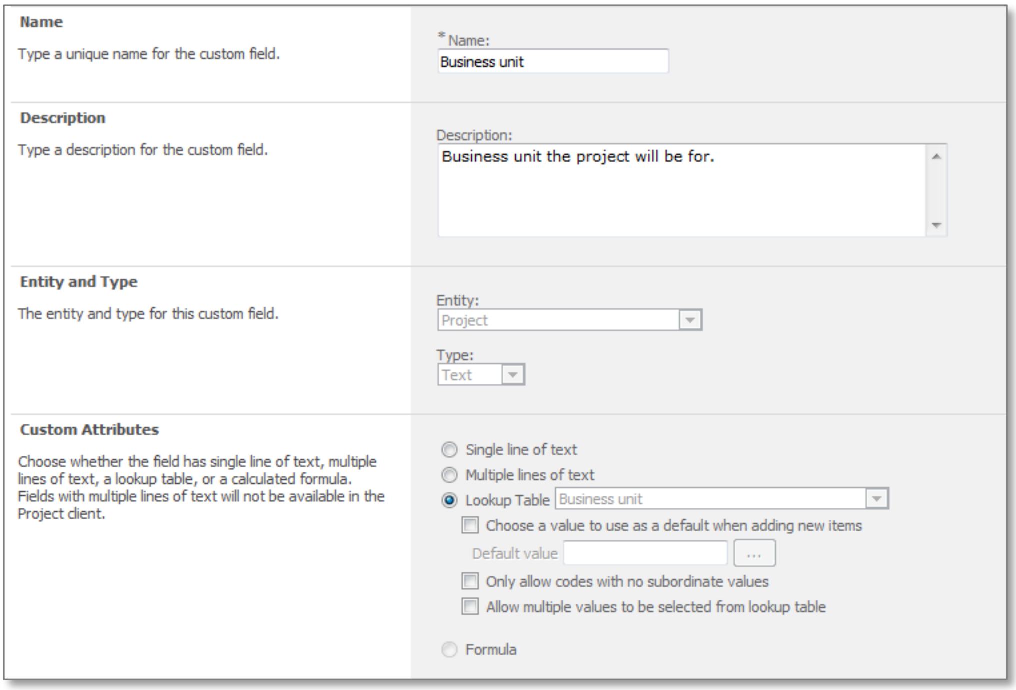
□ Use this field for matching generic resources

Figure 35: Position Role Custom Field

Finally, create a Project Level Custom Field called Business Unit.

a. In the **Name** field type, “Business unit”.
b. In the **Description** field type, “The business unit the project will be for”.
c. In the **Entity and Type** field, select the **Project** entity and type **Text**.
d. In the **Custom Attributes** section, select **Lookup Table** and choose the **Business unit** lookup table.

This field will be used in the project proposal for the initiator to record which business unit the project proposal is for.



**Name**

Type a unique name for the custom field.

* Name: Business unit

**Description**

Type a description for the custom field.

Description: Business unit the project will be for.

**Entity and Type**

The entity and type for this custom field.

Entity: Project

Type: Text

**Custom Attributes**

Choose whether the field has single line of text, multiple lines of text, a lookup table, or a calculated formula. Fields with multiple lines of text will not be available in the Project client.

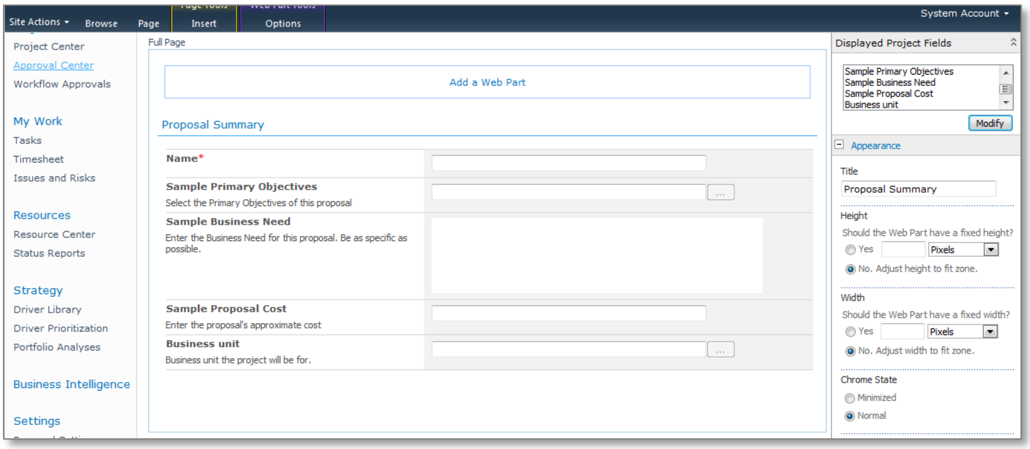
○ Single line of text
○ Multiple lines of text
● Lookup Table: Business unit
□ Choose a value to use as a default when adding new items
Default value: …
□ Only allow codes with no subordinate values
□ Allow multiple values to be selected from lookup table
○ Formula

Figure 36: Business Unit Custom Field

Note: Ensure the Business unit custom field is set to be workflow controlled.

## Project Detail Pages

Edit the **Proposal Summary** project detail page to add the Business unit field to the page.



Proposal Summary

**Name***

**Sample Primary Objectives**
Select the Primary Objectives of this proposal

**Sample Business Need**
Enter the Business Need for this proposal. Be as specific as possible.

**Sample Proposal Cost**
Enter the proposal's approximate cost

**Business unit**
Business unit the project will be for.

Displayed Project Fields
Sample Primary Objectives
Sample Business Need
Sample Proposal Cost
Business unit

Appearance

Title: Proposal Summary

Height
Should the Web Part have a fixed height?
○ Yes Pixels
● No. Adjust height to fit zone.

Width
Should the Web Part have a fixed width?
○ Yes Pixels
● No. Adjust width to fit zone.

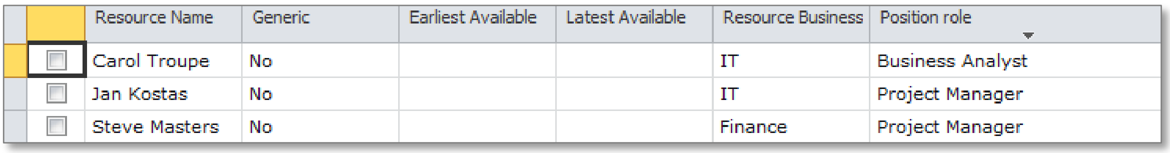
Chrome State
○ Minimized
● Normal

Figure 37: Modified Proposal Summary Project Detail Page

## Resources

Finally, create some Project Manager resources in the Resource centre (you may already have resources available in your Project Centre and can simply update the Business Unit and Position Role fields for that resource).

For this example, we will add three new resources, including configuring the custom fields we added, as below. These resources will be queried by the workflow to find the relevant Project Manager for the business unit chosen.



| | Resource Name | Generic | Earliest Available | Latest Available | Resource Business | Position role |
|---|---|---|---|---|---|---|
| □ | Carol Troupe | No | | | IT | Business Analyst |
| □ | Jan Kostas | No | | | IT | Project Manager |
| □ | Steve Masters | No | | | Finance | Project Manager |

Figure 38: Example Resources

## Building the Workflow

1. To create a Demand Management Workflow, select **Site Actions** > **Project Server Workflows** > **Create Demand Management workflow**.
2. Drag a **Set project stage** action onto the workflow canvas.

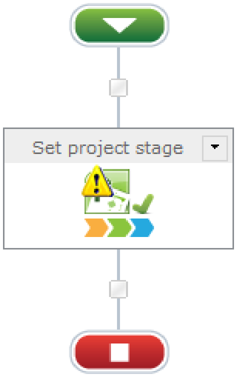
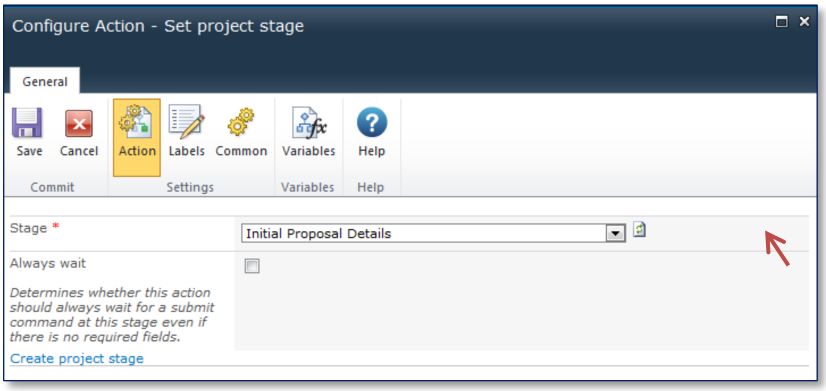


Figure 39: Project Stage Workflow Action

3. Double-click on the **Set project stage** action to configure.
   a. In the **Stage** field, select **Initial Proposal Details.**



Stage *: Initial Proposal Details

Always wait: □

*Determines whether this action should always wait for a submit command at this stage even if there is no required fields.*

Create project stage

Figure 40: Configure Project Stage

In this example, the workflow is required to move back and forward depending on the outcomes of various tasks and will need to leverage a **State Machine** action. The workflow can be configured to move back and forth between states by using a **Change State** action.

4. Drag a **State Machine** action onto the workflow canvas.

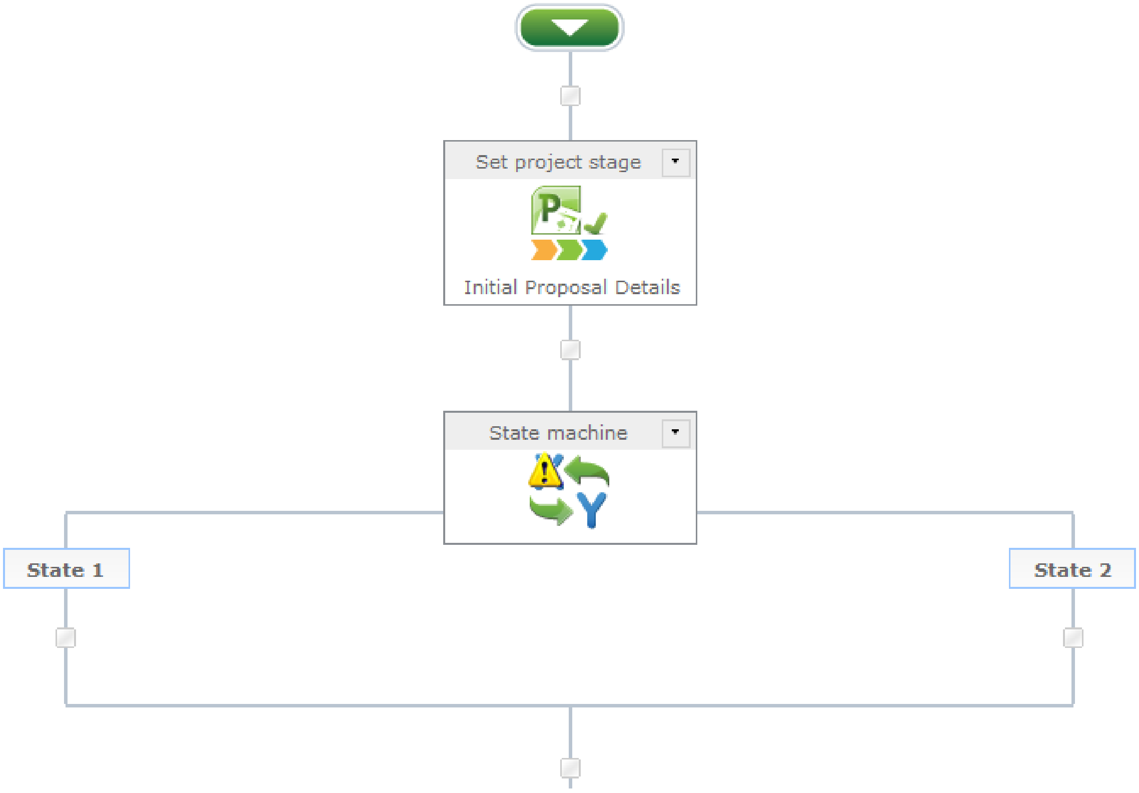
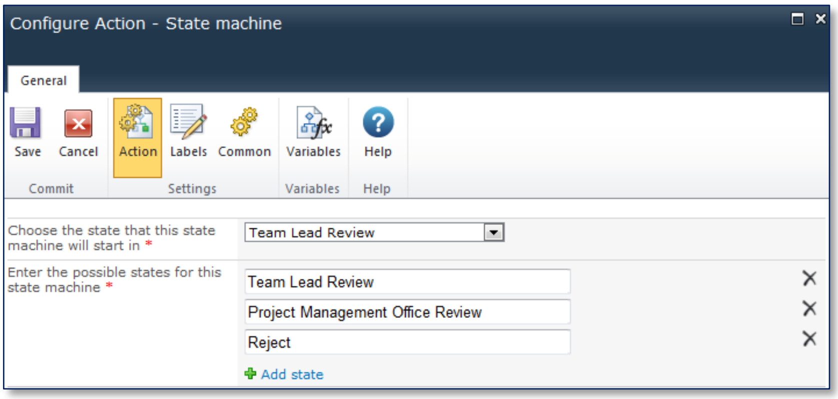


Figure 41: Adding an State Machine to the Design Canvas

5. By default, the state machine will have two states. Double-click on the **State Machine** action to configure.
   a. In the first state type, “Team Lead Review”.
   b. In the second state type, “Project Management Office Review”.
   c. Click **Add state** to add a third state. Type, “Reject”, as the title.
   d. In the **Choose the state that this state machine will start in** field, select **Team Lead Review.**



Choose the state that this state machine will start in * Team Lead Review

Enter the possible states for this state machine *
Team Lead Review
Project Management Office Review
Reject

Figure 42: Configure the State Machine Action

Next, drag a Project Server **Wait for Submit** action into the **Team Lead Review** state machine branch. This will ensure that each time the workflow enters that branch the project will have to be submitted, ensuring the Project Manager has an opportunity to control the resubmission of the proposal.

Once complete, the workflow should look as per **Figure 43: First State Completed**.

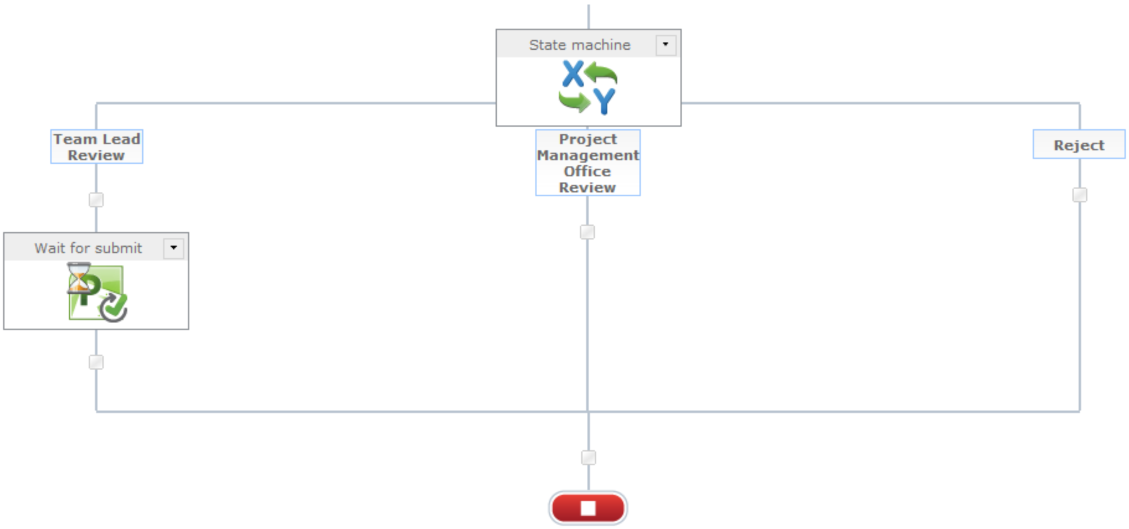


Figure 43: First State Completed

Configure the team lead review to determine the team lead users and to assign a flexi task.

6. Select a **Read project security group** action and an **Assign Flexi task** action and place on the Team Lead Review branch.

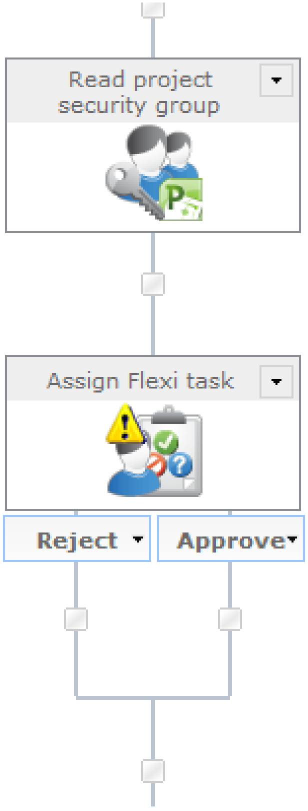
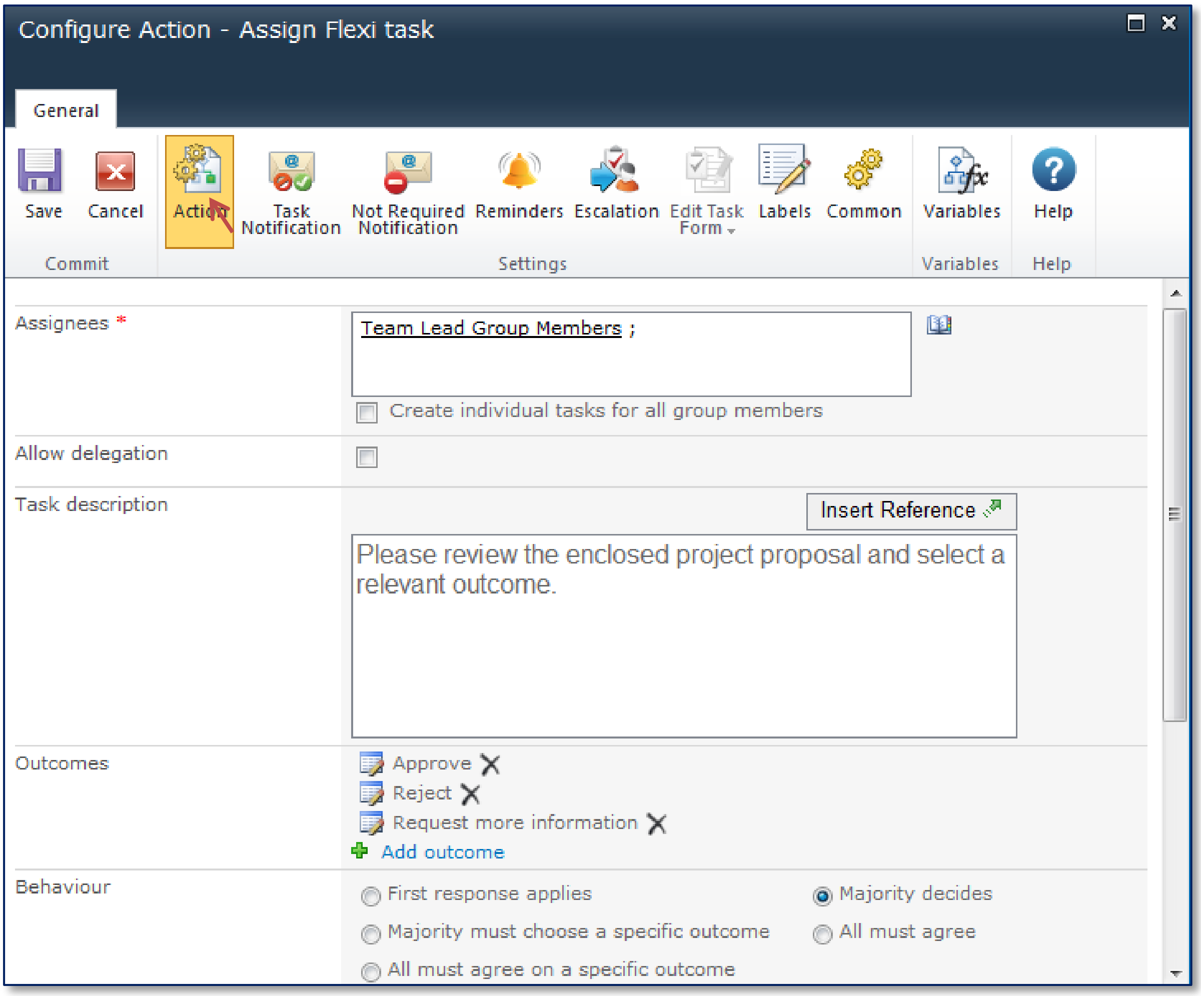


Figure 44: Add Read Project Server Security Group and Flexi Task to the Design Canvas

7. Double-click on the **Read project security group** action and configure.
    a. In the **Group Name** field, choose the 'Team Leads' Project Server Security group.
    b. Create a new single line of text variable called '**Team Lead Group Members'**. This will be used to store the members of the security group.
    c. In the **Store Result In** field, choose the **Team Lead Group Members** variable.

In this example, the **Assign Flexi task** will have three options: **Approve**, **Reject** and **Request More Information.**

8. Double-click on the **Assign Flexi Task** action to configure.
    a. In the **Assignees** field, click on the **Lookup icon** and select the **Team Lead Group Members** variable.
    b. In the **Task Description** type, “Please review the enclosed project proposal and select a relevant outcome”.
    c. Click **Add outcome,** and name the outcome, “Request more information”.
    d. In the **Behaviour** field, select **Majority decides.**



Assignees *: Team Lead Group Members ;

□ Create individual tasks for all group members

Allow delegation □

Task description: Please review the enclosed project proposal and select a relevant outcome.

Outcomes:
Approve
Reject
Request more information

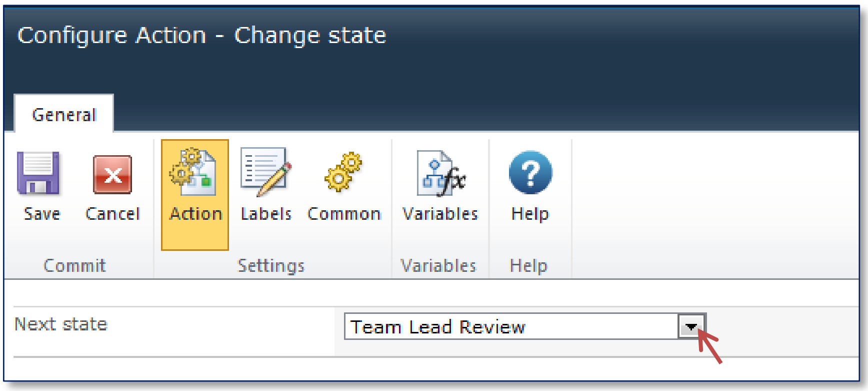
Behaviour:
○ First response applies ● Majority decides
○ Majority must choose a specific outcome ○ All must agree
○ All must agree on a specific outcome

Figure 45 : Configure Team Lead Flexi Task

    e. Click **Save**.

Different logic is required depending on the outcome chosen in the flexi task. If the outcome is approved, the workflow will change over to the PMO review stage; if more information is required, the workflow will start the team lead approval stage again, waiting for the Project manager to resubmit; and finally if the outcome is rejected, the workflow will swap over to the rejected state.

9. Add a **Change State** action onto each of the three flexi task outcome branches. Each **Change state** action needs to be configured to set the next state.



Next state: Team Lead Review

Figure 46 : Configure Change State Action

10. Configure the states as follows:

| Flexi Task Outcome | Next state setting |
|---|---|
| Reject | Reject |
| Approve | Project Management Office Review |
| Request more information | Team Lead Review |

Once completed the workflow should look like this:

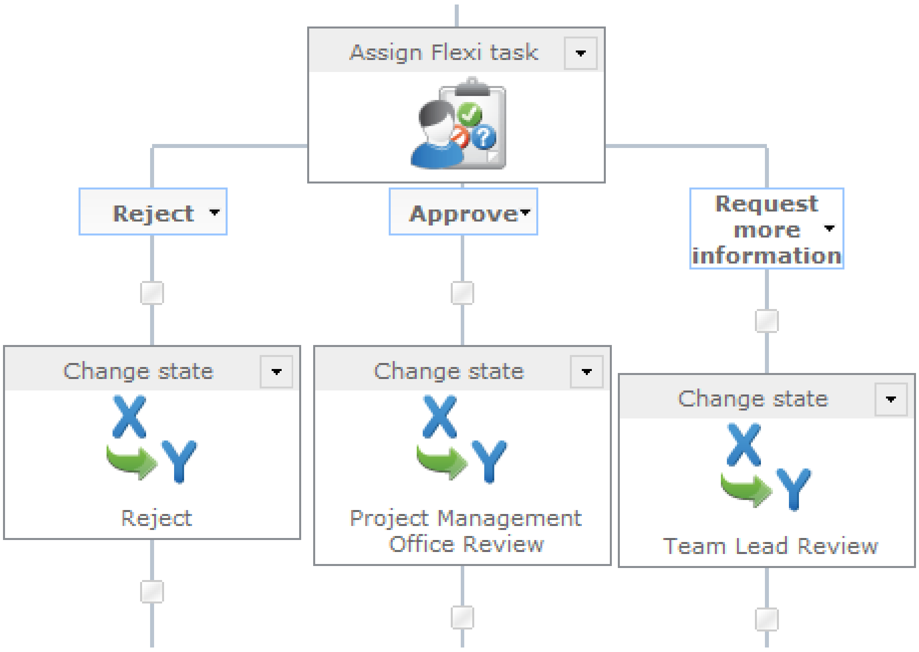


Figure 47 : Configured Flexi Task and Change States

The next step is to configure the **Project Management Office Review** state branch. This branch will be similar to the Team Lead Review, but will use the Portfolio Manager Security Group to determine which users will receive the approvals.

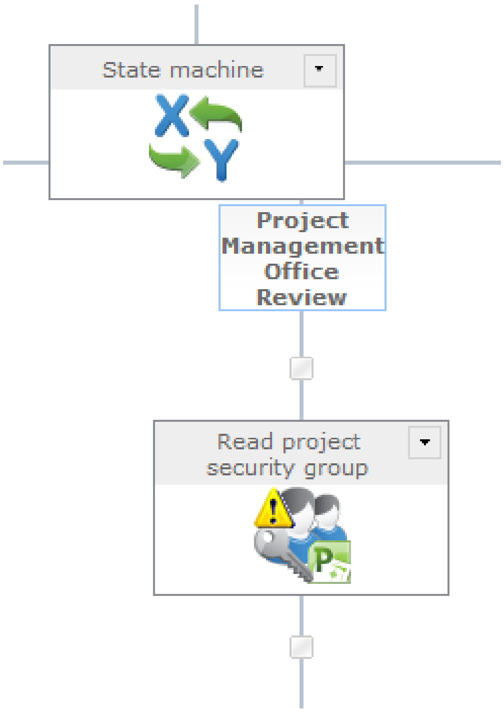


Figure 48 : PMO Review State Branch

11. Double-click on the **Read project security group** action.
    a. Configure the **Group Name** to read, “Portfolio Managers”.
    b. Create a new single line of text variable called ‘**Portfolio Manager Group Members’**. This variable will be used in the **Flexi Task** later in the workflow.
    c. In the **Store result in** field, select ‘Portfolio Manager Group Members’.

12. Next add an **Assign Flexi task** action to the workflow.

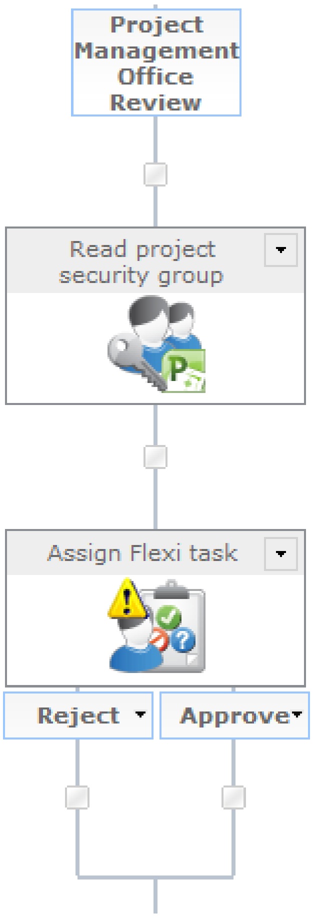


Figure 49 : PMO Review Flexi Task

13. Double-click on the **Assign Flexi Task** action to configure.
    f. In the **Assignees** field, click on the **Lookup icon** and select the **Portfolio Manager Group Members** variable.
    g. In the **Task Description** type, "Please review the enclosed project proposal and select a relevant outcome".
    h. Click **Add outcome,** and name the outcome, "Request more information".
    i. In the **Behaviour** field, select **Majority decides.**

14. Configure the three outcome branches; Reject, Approve and Request more information.
    - For the **Reject** outcome, add a **Change State** action and change the state to **Rejected.**
    - For the **Request more information** outcome, add a **Change State** action and change the state to **Team Lead Review**. This will cause the approval loop to restart and require the project manager to resubmit the project into the workflow.

Once completed, the workflow should look like the workflow in **Figure 50 : Configured Flexi** Task.

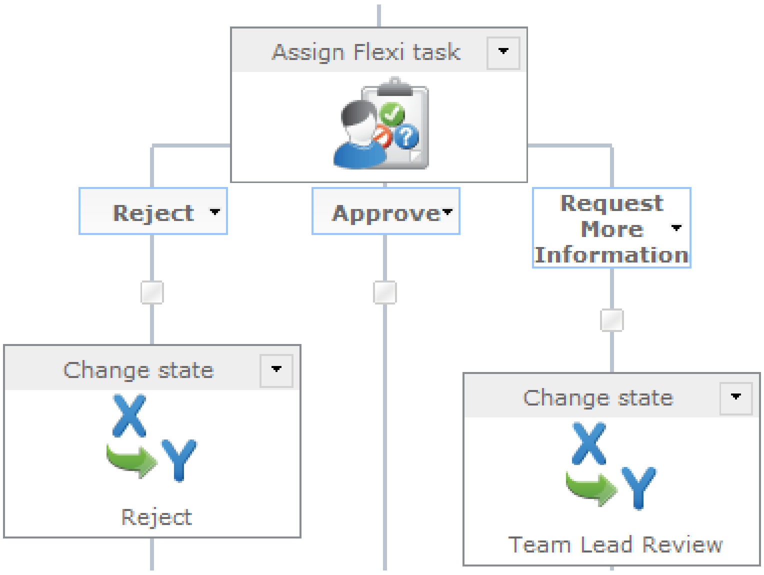


Figure 50 : Configured Flexi Task

15. To configure the Reject state of the state machine, add a **Set project stage** action and an **End workflow** action to the branch as follows:

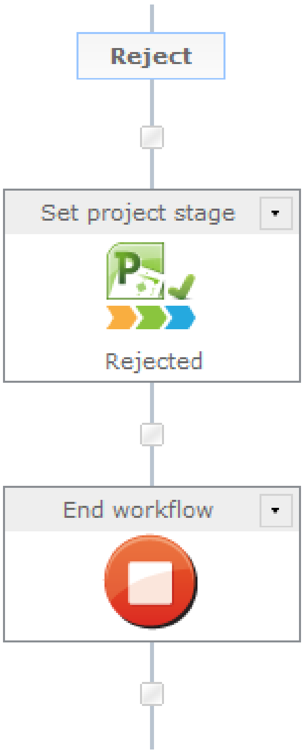


Figure 51 : Rejected Branch

Configure the **Set project stage** action to change the project to **Rejected.**

Finally, it is important to tell the workflow that the state machine is complete.

16. Add a **Change state** action to the Approved outcome of the flexi task. The completed PMO review flexi task workflow should look like the workflow in **Figure 12 : Completed PMO Review Flexi** Task**.**

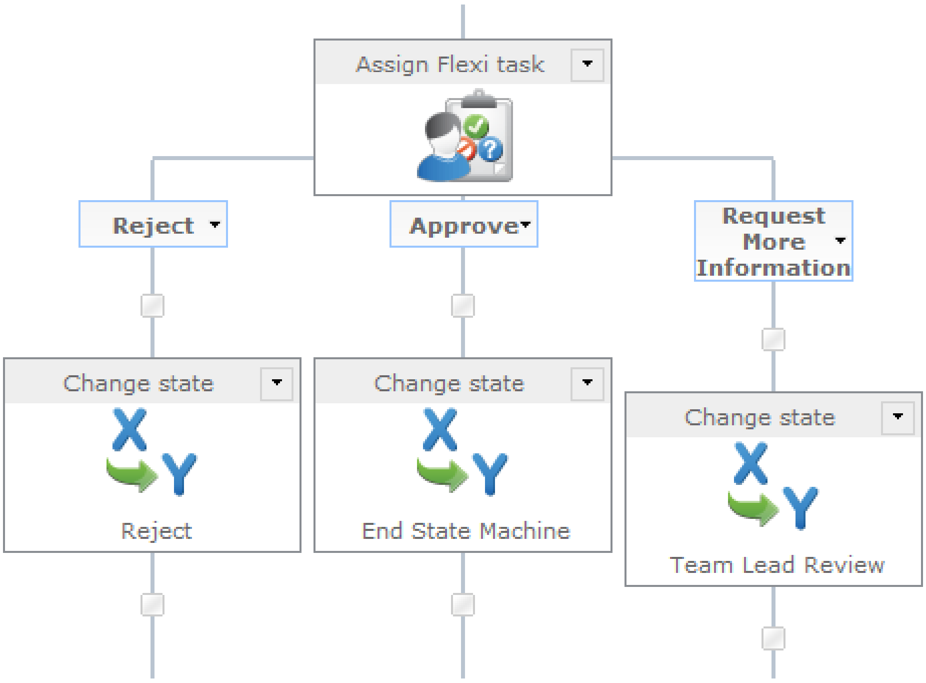


Figure 12 : Completed PMO Review Flexi Task

## Determine the Project Manager for the Business Unit

Once the approvals have been completed, the next step in the workflow is to determine the correct Project Manager for the business unit so they can be assigned as the project owner for delivery. Previously this had been a manual process, requiring a member of the PMO to determine the correct Project Manager and then opening up the project schedule and assigning that person as the project owner.

With Nintex Workflow for Project Server, it is now possible to leverage the new **Query Project Server** action, which allows access to all the PSI Read methods in Project Server.

In this example, we will use these actions to:

- Determine the custom field GUID of the 'Project Manager' position role
- Determine the custom field GUID of the 'Business Unit' entered in the proposal
- Retrieve the unique id's of any resources that have the relevant custom fields

As the first two actions can be performed independently of one another, we can make use of the '**Run Parallel Actions'** action.

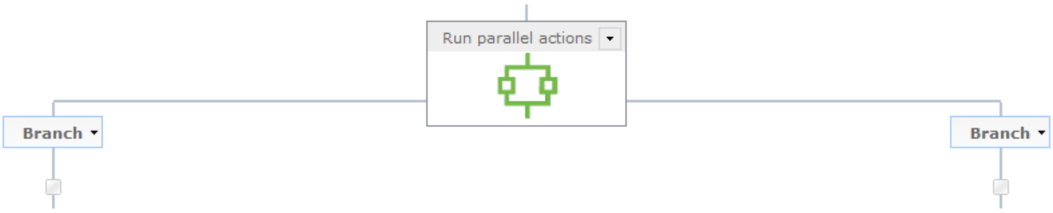


Figure 53: Run Parallel Actions

1. Select a **Run Parallel** workflow action and place on the workflow canvas.
2. On the left-hand branch of the **Run parallel** action, drag two **Query project server** actions. These will be used to determine the ID of the 'Project Manager' **Position Role.**

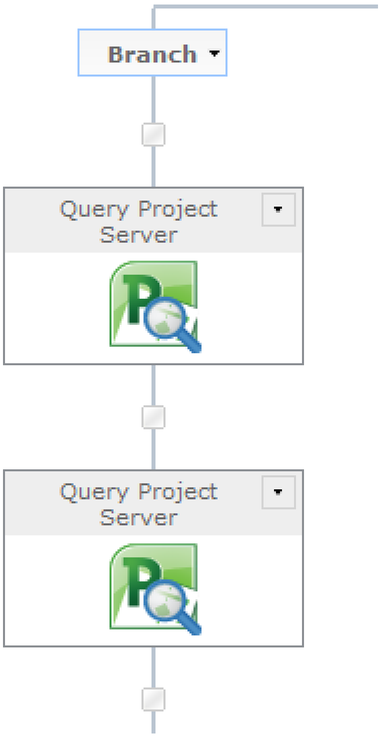


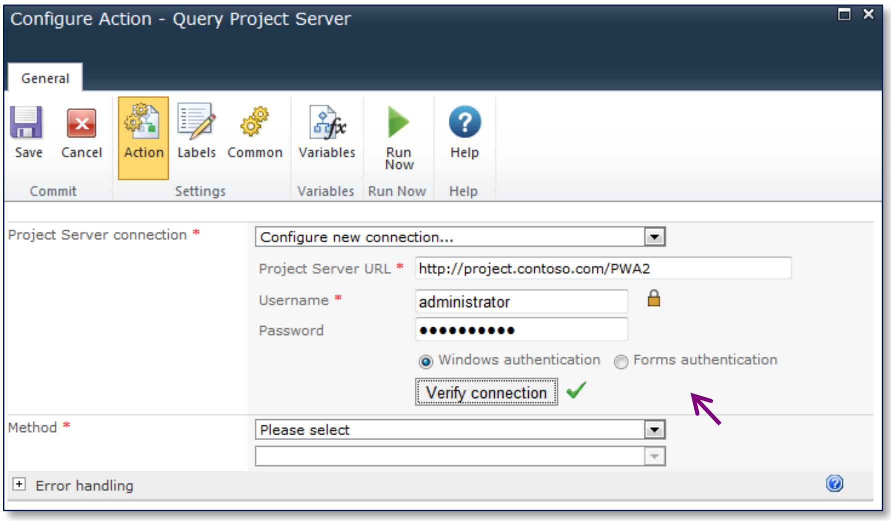
Figure 13 : Left Hand Branch Configuration

As the '**Query Project Server'** action needs to communicate with Project Server through the PSI, it is necessary to provide credentials for the action before any of the available methods can be chosen.

There are two mechanisms to enter credentials, either directly into the action (see screenshot below), or they can be configured by the workflow administrator centrally within the Nintex Workflow for Project Server Central Administration. This allows the administrator to enter the credentials in one location and the workflow designer will not need to repeat them.

*Note: For more information on setting up the PWA credentials, refer to Appendix A.*

3. Once the credentials have been chosen or entered, click **Verify connection**.



Project Server connection * Configure new connection...

Project Server URL * http://project.contoso.com/PWA2

Username * administrator

Password ●●●●●●●●●●

● Windows authentication ○ Forms authentication

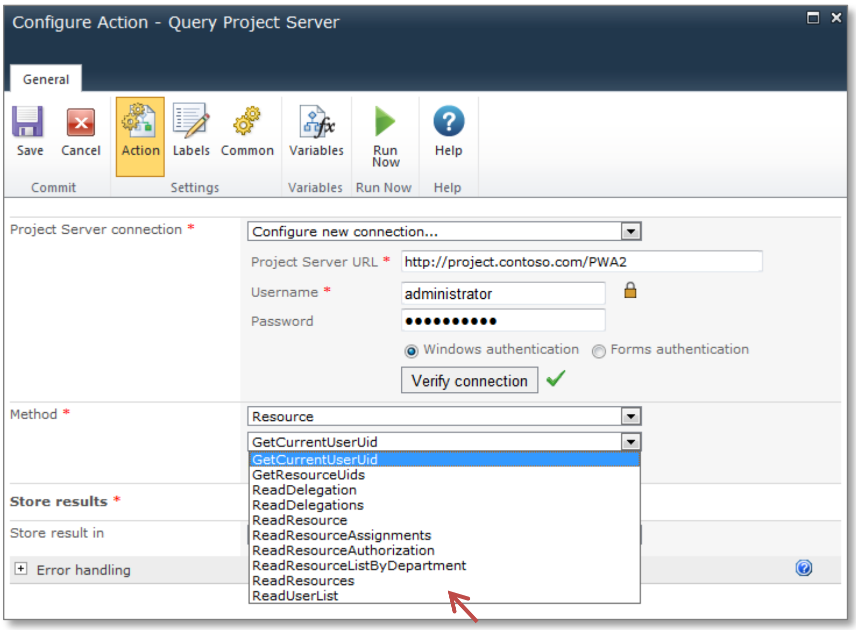
Verify connection ✓

Method * Please select

⊞ Error handling

Figure 55 : Query Project Server Action

If the credentials entered are correct, a tick will be displayed and the method drop down list will become populated; allowing you to choose the relevant method to query the data from the PSI.



Project Server connection * Configure new connection...

Project Server URL * http://project.contoso.com/PWA2

Username * administrator

Password ●●●●●●●●●●

● Windows authentication ○ Forms authentication

Verify connection ✓

Method * Resource

GetCurrentUserUid

- GetCurrentUserUid
- GetResourceUids
- ReadDelegation
- ReadDelegations
- ReadResource
- ReadResourceAssignments
- ReadResourceAuthorization
- ReadResourceListByDepartment
- ReadResources
- ReadUserList

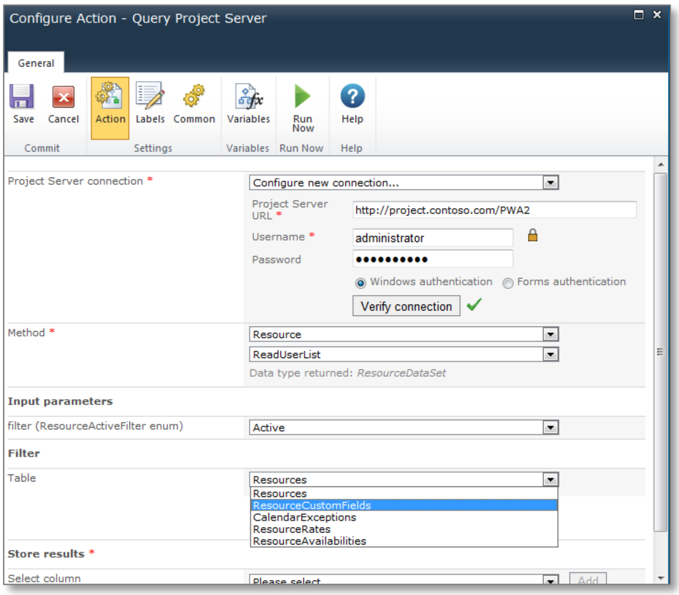
**Store results** *

Store result in

⊞ Error handling

Figure 56 : Select PSI Method

Once the method is chosen, the action will automatically show the input parameters required to use the query. You can then select the table returned from Project Server that you wish to use in your workflow.



Configure Action - Query Project Server

Project Server connection * : Configure new connection...

Project Server URL * : http://project.contoso.com/PWA2

Username * : administrator

Password : ••••••••••

● Windows authentication ○ Forms authentication

Verify connection ✓

Method * : Resource / ReadUserList

Data type returned: *ResourceDataSet*

**Input parameters**

filter (ResourceActiveFilter enum) : Active

**Filter**

Table : Resources

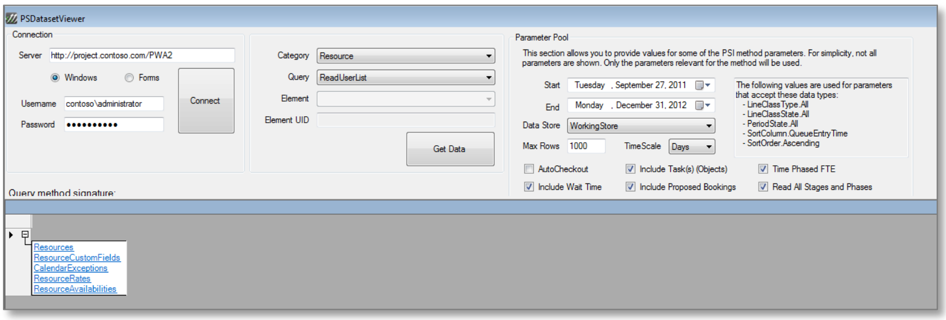
Resources
ResourceCustomFields
CalendarExceptions
ResourceRates
ResourceAvailabilities

**Store results** *

Select column : Please select

Figure 57 : Possible Tables Returned by the Query

Note: Nintex provides a handy tool called the **PSDataSetViewer** that allows you test the query methods against your Project Server instance, selecting the relevant parameters and viewing the various data sets that are returned. This tool can be invaluable when trying to determine which query method to use. The **PSDataSetViewer** is available from http://connect.nintex.com for registered users of Nintex Workflow for Project Server.



PSDatasetViewer

Connection

Server : http://project.contoso.com/PWA2

● Windows ○ Forms

Username : contoso\administrator

Password : ••••••••••

Category : Resource

Query : ReadUserList

Element :

Element UID :

Parameter Pool

This section allows you to provide values for some of the PSI method parameters. For simplicity, not all parameters are shown. Only the parameters relevant for the method will be used.

Start : Tuesday , September 27, 2011

End : Monday , December 31, 2012

Data Store : WorkingStore

Max Rows : 1000   TimeScale : Days

The following values are used for parameters that accept these data types:
- LineClassType.All
- LineClassState.All
- PeriodState.All
- SortColumn.QueueEntryTime
- SortOrder.Ascending

□ AutoCheckout ☑ Include Task(s) (Objects) ☑ Time Phased FTE

☑ Include Wait Time ☑ Include Proposed Bookings ☑ Read All Stages and Phases

Query method signature:

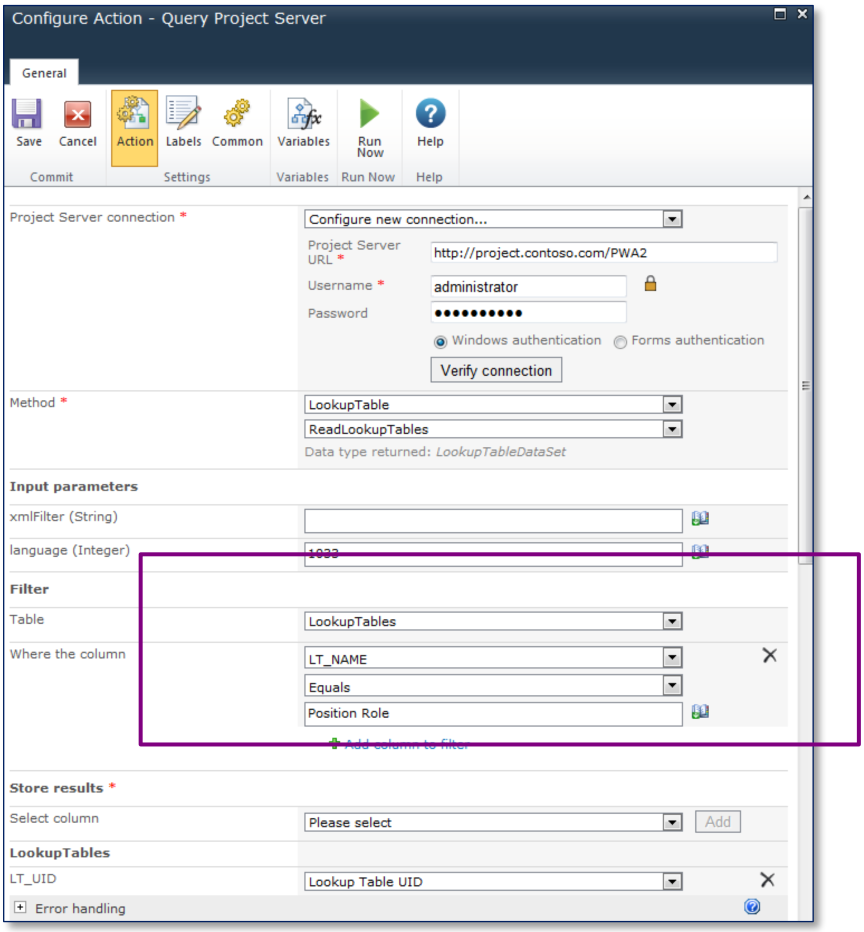
Resources
ResourceCustomFields
CalendarExceptions
ResourceRates
ResourceAvailabilities

Figure 58 : PSDataSetViewer Tool

The first **Query project server** action will be configured to retrieve the UID of the Position Role Lookup Table and store it in a new variable called **LookupTable_UID** as outlined below:

3. In the **Method** field:
   a. Select the menu to choose the group to call. Select **Lookup Tables**.
   b. In the second menu, choose the **ReadLookupTables** method.

4. In the language input parameter, enter the language code of 1033.



Project Server connection * : Configure new connection...

Project Server URL * : http://project.contoso.com/PWA2

Username * : administrator

Password : ●●●●●●●●●●

● Windows authentication ○ Forms authentication

Method * : LookupTable / ReadLookupTables

Data type returned: *LookupTableDataSet*

**Input parameters**

xmlFilter (String) :

language (Integer) : 1033

**Filter**

Table : LookupTables

Where the column : LT_NAME / Equals / Position Role

**Store results** *

Select column : Please select

**LookupTables**

LT_UID : Lookup Table UID

Figure 59 : Query Lookup Table UID

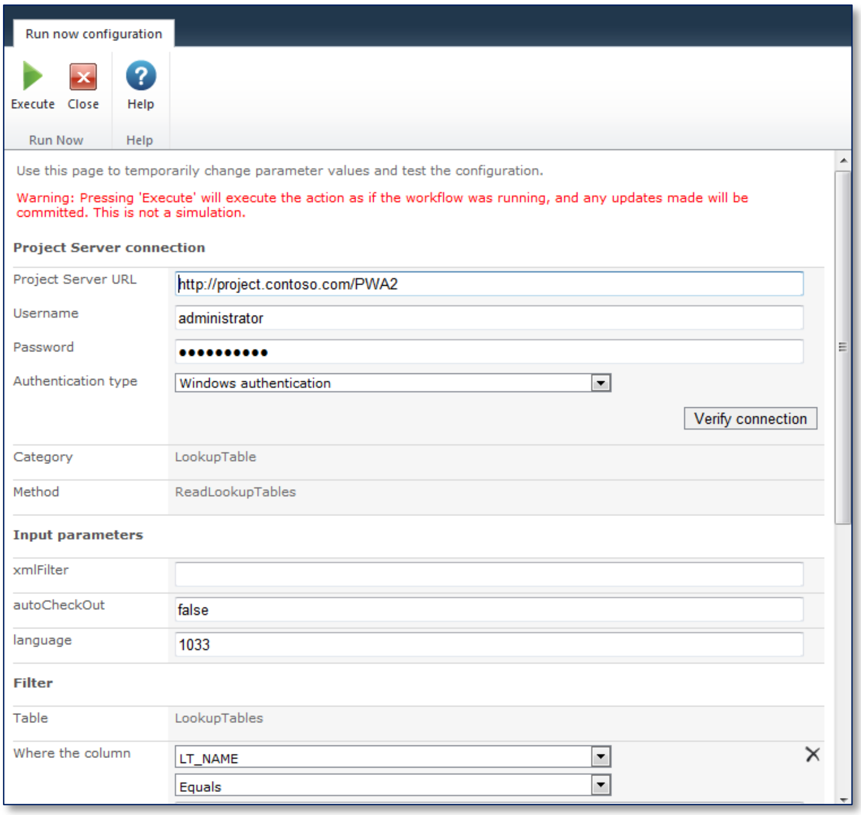
5. The Query Project Server action returns all the associated data tables for the method called. In this example, we are only interested in certain items in the **LookupTables** table, so it is necessary to set up a filter:
   a. In the Table menu item, select **LookupTables.**
   b. Click on the **Add Column to filter** link.
   c. For the **Where the column** area, select **LT_NAME Equals** and enter the text **Position Role.**

6. Finally, select **LT_UID** from the **LookupTable** group in the **Select Column Field** and choose to store the value in a single line of text variable called **Lookup Table UID.**

Before moving onto the next action, it is worth checking to confirm if the action has been configured correctly.

With Nintex Workflow 2.3 or higher, a new command, **Run Now,** has been made available to test certain types of actions which interact with external data. When clicked, a new window will appear allowing the action to be run.

*Note: Once Run Now has been selected, it will execute the action as if the workflow was running and any updates made will be committed.*



Use this page to temporarily change parameter values and test the configuration.

Warning: Pressing 'Execute' will execute the action as if the workflow was running, and any updates made will be committed. This is not a simulation.

**Project Server connection**

| | |
|---|---|
| Project Server URL | http://project.contoso.com/PWA2 |
| Username | administrator |
| Password | •••••••••• |
| Authentication type | Windows authentication |
| Category | LookupTable |
| Method | ReadLookupTables |

**Input parameters**

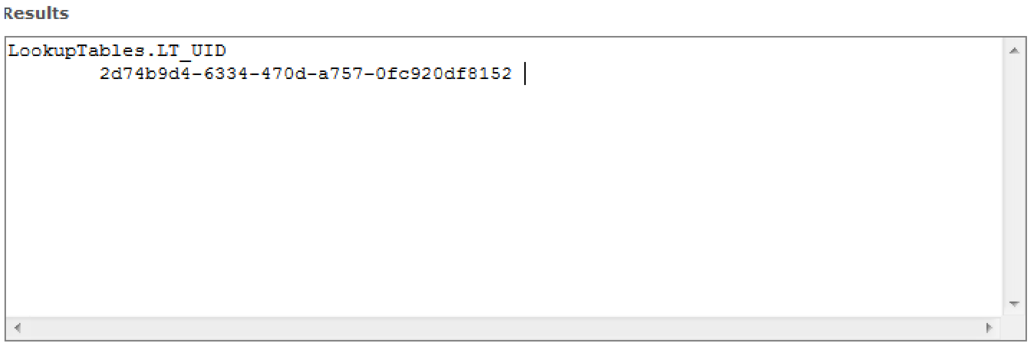
| | |
|---|---|
| xmlFilter | |
| autoCheckOut | false |
| language | 1033 |

**Filter**

| | |
|---|---|
| Table | LookupTables |
| Where the column | LT_NAME<br>Equals |

Figure 60 : Checking the Query With ‘Run Now’

If everything has been configured correctly, we should see one GUID returned for the Position role lookup table. This can be verified using the **PSDataSetViewer** application if required.



```
LookupTables.LT_UID
        2d74b9d4-6334-470d-a757-0fc920df8152
```

Figure 61 : Run Now Results

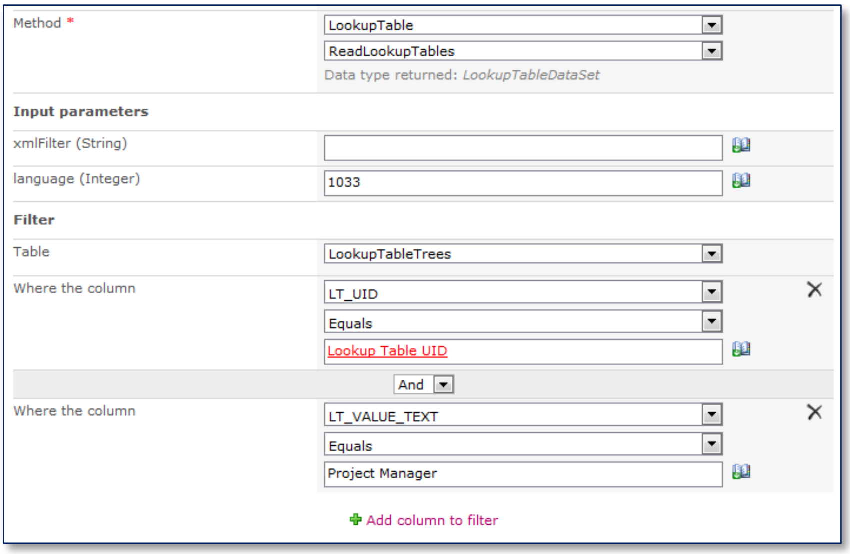
7. Click **Close** and then **Save** to save the **Query project server** configuration.

The second **Query project server** action will take the **Lookup Table UID** determined in the previous action and perform another lookup using the **ReadLookupTables** method to determine the UID of the Project Manager role.

8. In the **Method** field:
    a. Select the menu to choose the group to call. Select **LookupTables**.
    b. In the second menu, choose the **ReadLookupTables** method.

9. In the **language input** parameter, enter the language code of 1033**.**

10. In the Filter area, select:
    a. In the **Table** menu, **LookupTableTrees.**
    b. Click on **Add column to Filter**, in the **Where the column** area**,** select **LT_UID Equals** and enter the variable **Lookup Table UID.**
    c. Choose **Add column to Filter** to add a second filter.
    d. In the **Where the column** area**,** select **LT_VALUE_Text Equals** and enter the text **Project Manager.**



Method * LookupTable
ReadLookupTables
Data type returned: *LookupTableDataSet*

**Input parameters**

xmlFilter (String)

language (Integer) 1033

**Filter**

Table LookupTableTrees

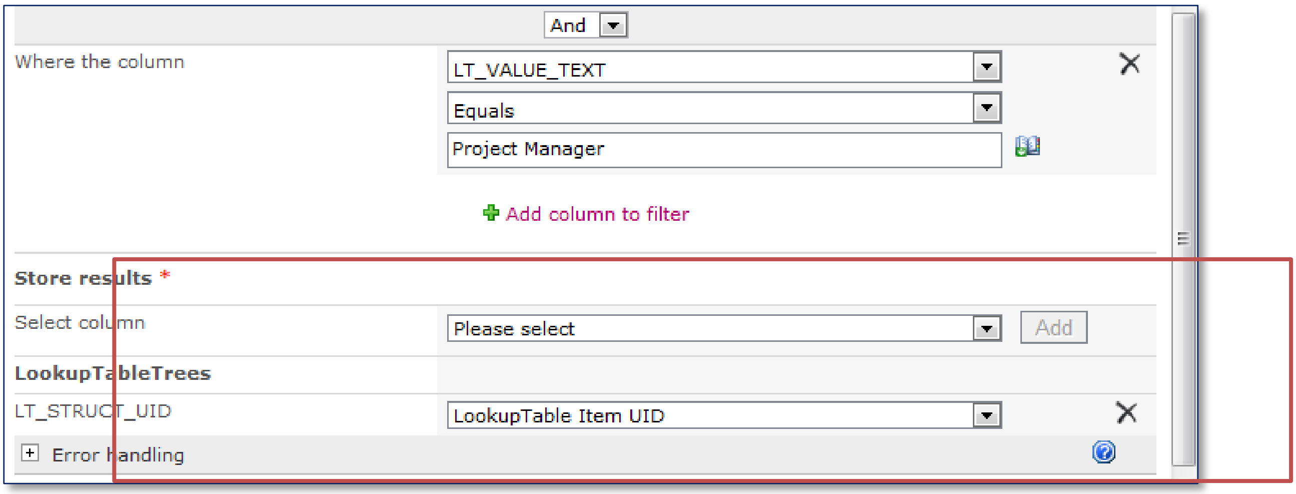
Where the column LT_UID
Equals
Lookup Table UID

And

Where the column LT_VALUE_TEXT
Equals
Project Manager

Figure 62 : Second Query Project Server Filter Settings

11. Finally, select the column **LT_STRUCT_UID** and choose to store it in a new single line of test variable called **LookupTable Item UID**.



And

Where the column LT_VALUE_TEXT
Equals
Project Manager

**Store results** *

Select column Please select

**LookupTableTrees**

LT_STRUCT_UID LookupTable Item UID

Figure 63 : Determining the Project Manager Lookup Table Item UID

The right-hand branch of the parallel action will be used to retrieve the business unit entered by the user and then use **Query project server** actions to retrieve the relevant GUID of the business unit custom field.

1. Select a **Read project property** action and two **Query project server** actions and place on the right-hand branch of the **Run parallel** action as per **Figure 64 : Right Hand Branch** Configuration.

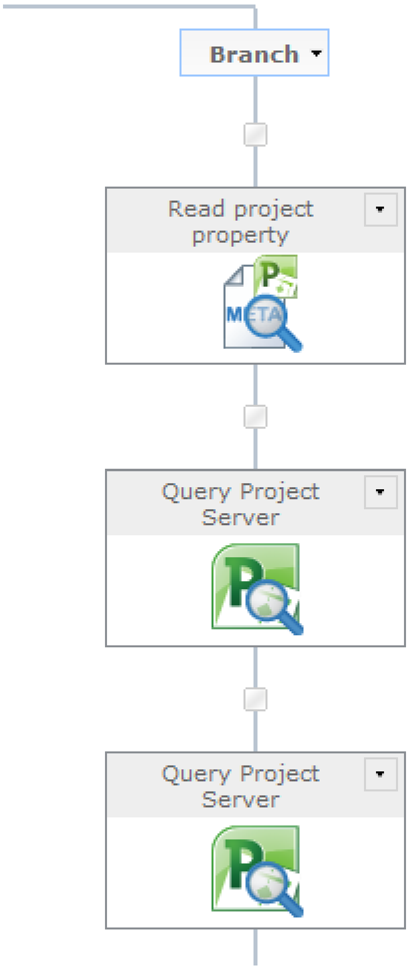
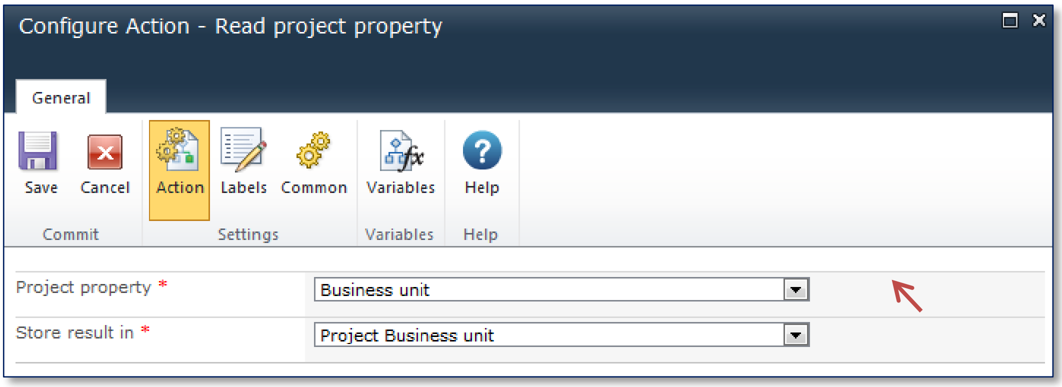


Figure 64 : Right Hand Branch Configuration

2. Configure the **Read project property** action.
   a. In the **Project Property** field, select **Business unit.**
   b. Create a new single line of text variable called **Project Business unit** and select it in the **store result in** menu.

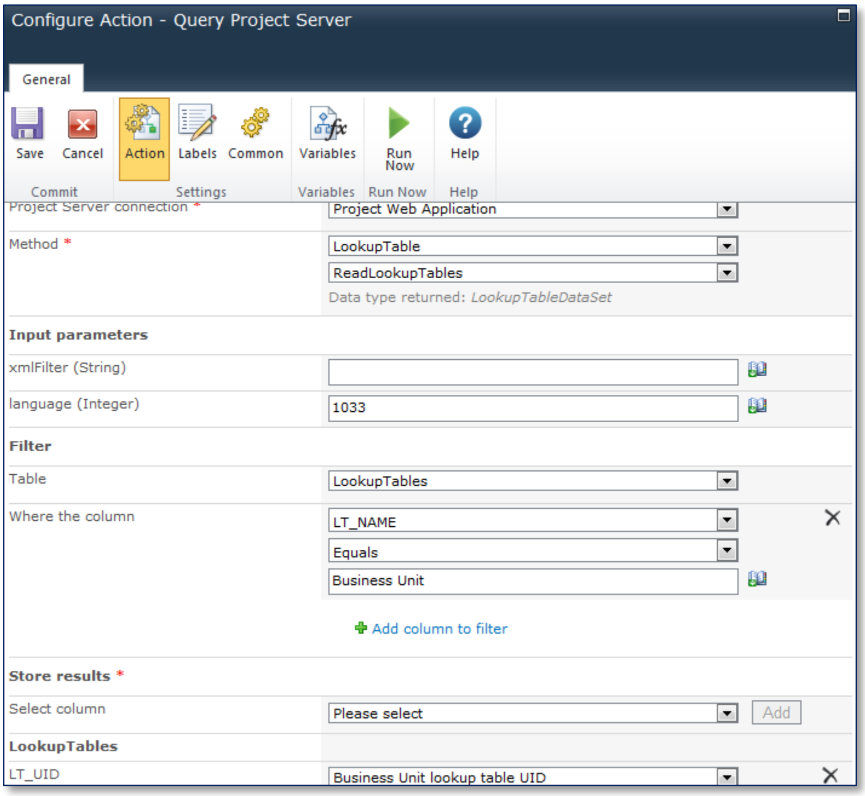


Project property * Business unit

Store result in * Project Business unit

Figure 65 : Read Business Unit Project Property

3. Next, configure the **Query project server** actions to determine the 'Business unit' lookup table and to then retrieve the specific lookup table item for the business unit chosen. To do this:
    a. Select **LookupTable** in the first method menu.
    b. Select **ReadLookupTables** in the second method menu.
    c. Enter **1033** for the **language** input parameter.
    d. Select **LookupTables** in the table menu of the **Filter** area.
    e. Click on **Add column to Filter**, select **LT_NAME Equals** and enter the text **Business Unit** into the **Where the column** fields.
    f. Finally, select the column **LT_UID** from the **LookupTables** area in the results and save it in a new  single line of text variable called '**Business Unit Lookup Table UID**'.



Project Server connection * : Project Web Application

Method * : LookupTable
ReadLookupTables
Data type returned: *LookupTableDataSet*

**Input parameters**

xmlFilter (String) :

language (Integer) : 1033

**Filter**

Table : LookupTables

Where the column : LT_NAME
Equals
Business Unit

Add column to filter

**Store results** *

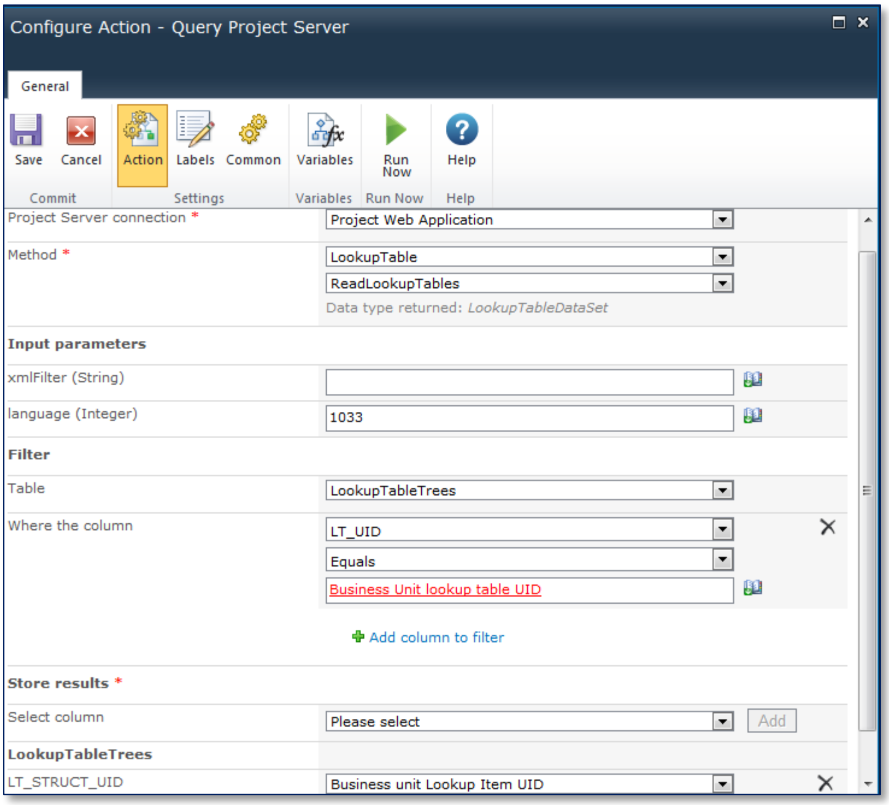
Select column : Please select — Add

**LookupTables**

LT_UID : Business Unit lookup table UID

Figure 66 : Query Read Lookup Tables Action

4. Configure the second Query Project Server action to retrieve the ID of the Business Unit, to do so:
    a. Select **LookupTable** in the first method menu.
    b. Select **ReadLookupTables** in the second method menu.
    c. Enter **1033** for the **language** input parameter.
    d. Select **LookupTableTrees** in the table menu of the **Filter** area.
    e. Click on **Add column to Filter**, select **LT_UID Equals** and **Business Unit Lookup table UID** into the **Where the column** fields.
    f. Finally, select the column **LT_STRUCT_UID** in the results and save it in a new single line of text variable called '**Business Unit Lookup Item UID**'.



Project Server connection * Project Web Application

Method * LookupTable
ReadLookupTables
Data type returned: *LookupTableDataSet*

**Input parameters**

xmlFilter (String)

language (Integer) 1033

**Filter**

Table LookupTableTrees

Where the column LT_UID
Equals
Business Unit lookup table UID

Add column to filter

**Store results** *

Select column Please select Add

**LookupTableTrees**

LT_STRUCT_UID Business unit Lookup Item UID

Figure 67 : Query Business Unit Lookup

Due to the way the custom fields are stored in Project Server, it isn't possible to retrieve the resources in one query, instead it is necessary to pull back all the resources that are Project Managers, then iterate through them, checking to see if that resource is in the business unit entered by the user.

5. To retrieve the resource information, add a **Query Project Server** action to the workflow that will use the **ReadResources** method and store the output in a collection variable called **Project Manager Resource Collection**. A collection is an array that will store items of the same data type; in this case, the GUIDs of all the Resources that are Project Managers.

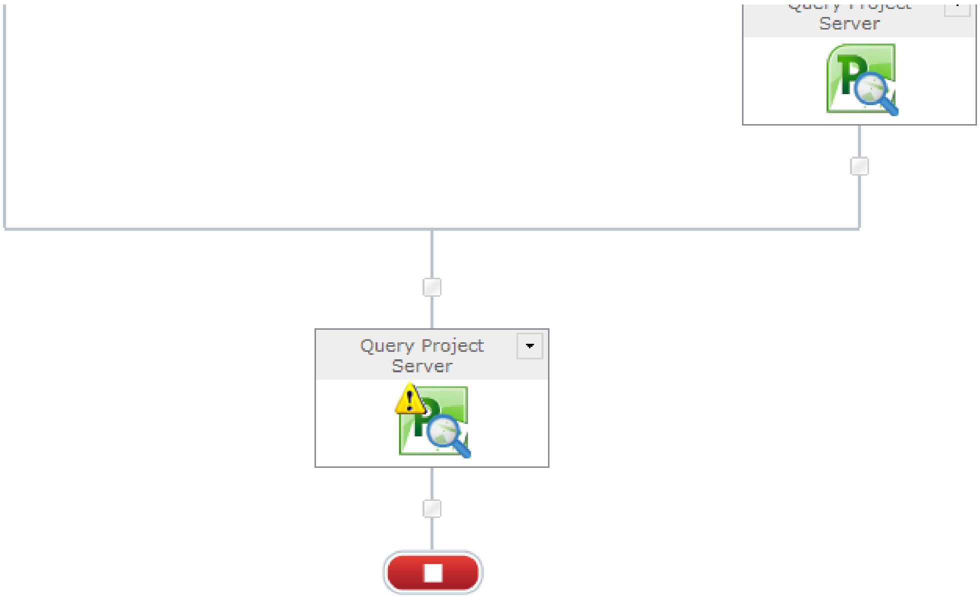


Figure 68 : Retrieve Resources Query Project Server Action

To do so, add a new Query Project Server action to the workflow and configure it as follows:

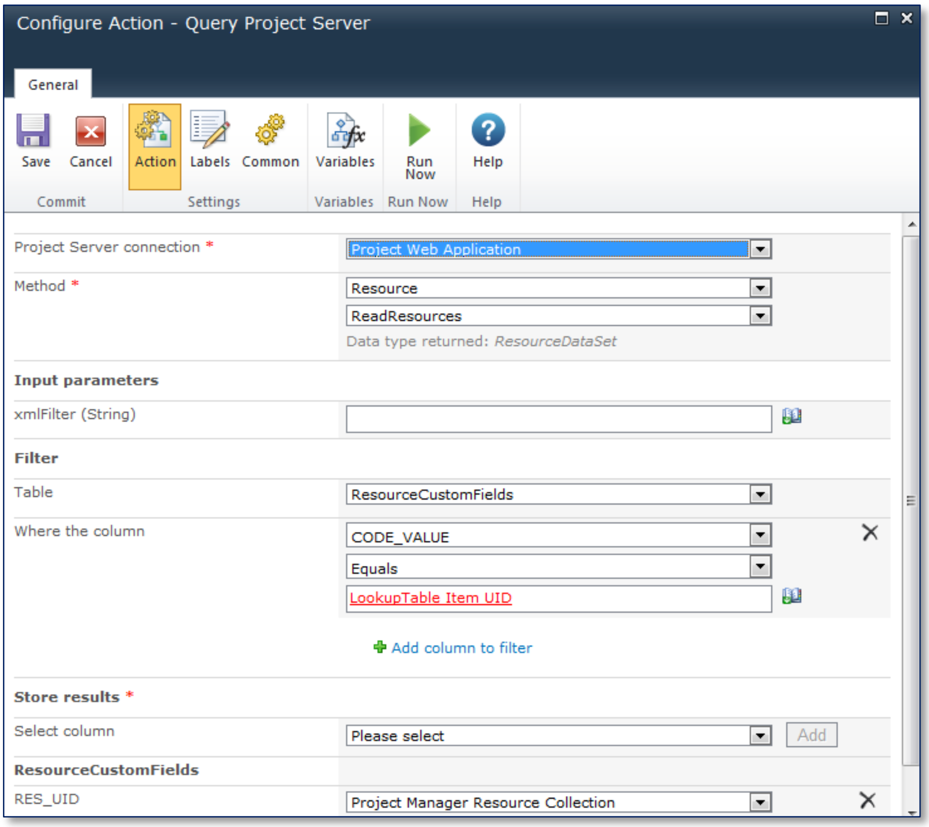
a. Select **Resource** in the first method menu.

b. Select **ReadResources** in the second method menu.

c. Select **ResourceCustomFields** in the table menu of the **Filter** area.

d. Click on **Add column to filter** and select **CODE_VALUE Equals** and select the variable **LookupTable Item UID** into the **Where the column** fields.

e. Finally, select the column **RES_UID** in the results and save it in a **collection** variable called ‘**Project Manager Resource Collection**’.



Configure Action - Query Project Server

| | |
|---|---|
| Project Server connection * | Project Web Application |
| Method * | Resource<br>ReadResources<br>Data type returned: *ResourceDataSet* |
| **Input parameters** | |
| xmlFilter (String) | |
| **Filter** | |
| Table | ResourceCustomFields |
| Where the column | CODE_VALUE<br>Equals<br>LookupTable Item UID |
| | Add column to filter |
| **Store results** * | |
| Select column | Please select   Add |
| **ResourceCustomFields** | |
| RES_UID | Project Manager Resource Collection |

Figure 69 : Query Resources into Resource Collection

To iterate through the **Project Manager Resource Collection**, add a **For each** action to the workflow, this enables the workflow to iterate through the collection one Resource ID at a time. Inside the **For each** action loop, add another **Query project server** action. This will be used to do a lookup of the resource and to determine if they are part of the business unit.

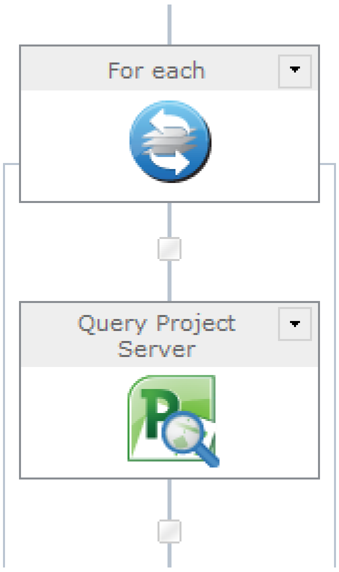
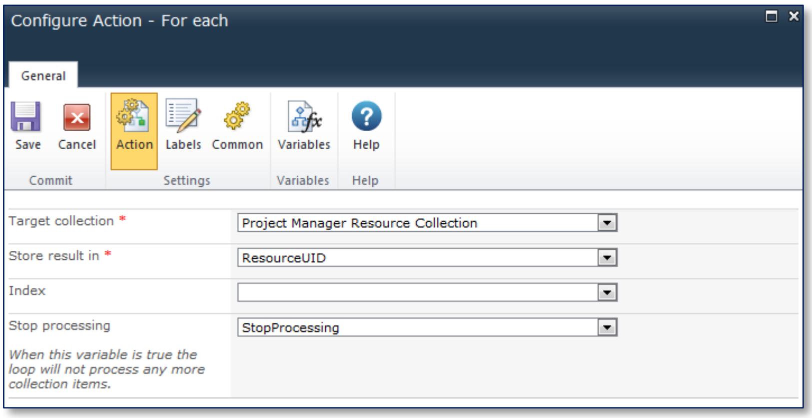


Figure 70 : For Each Action

1. Configure the **For each** action.
   a. In the **Target collection** field**,** select **Project Manager Resource Collection.**
   b. In the **Store Result in** field, create a new single line of text variable called **Resource UID** and then select it.
   c. In the **Stop processing** field, create a new yes/no variable called **StopProcessing** and then select it. By default Yes / No variables will default to false.



Target collection * Project Manager Resource Collection

Store result in * ResourceUID

Index

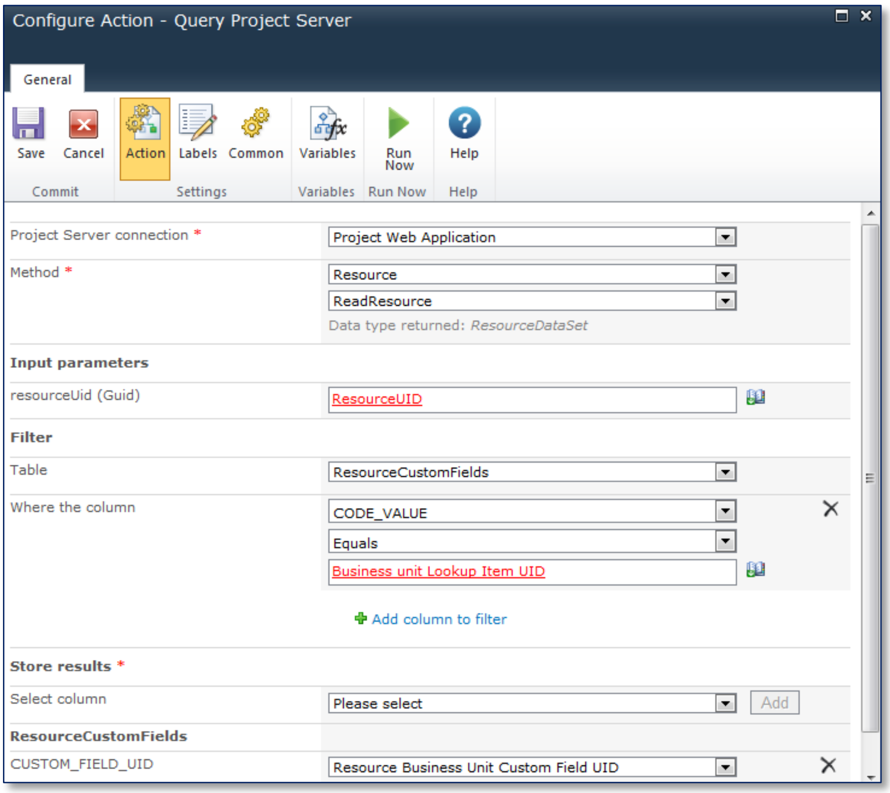
Stop processing StopProcessing

*When this variable is true the loop will not process any more collection items.*

Figure 71 : For Each Configuration

The action takes the **Project Manager Resource Collection** and loops around extracting a resource into the **ResourceUID** single line of text custom field until either all the items in the collection have been processed, or the Boolean variable **StopProcessing** is set to True.

2. Next, configure the **Query project server** action.
   a. In the **Method** field, select **Resource; ReadResource.**
   b. In the **ResourceUID(GUID)** field, use the **Lookup icon** to locate the **ResourceUID.**
   c. In the **Filter; table** field, select **ResourceCustomFields.**
   d. Select **Add Column to Filter** in the **Where the column** field, select **CODE_VALUE; Equals; Business Unit Lookup Item UID.**
   e. In the **CUSTOM_FIELD_UID** field, create a new single line of text variable called **Resource Business Unit Custom Field UID** and select it.



Project Server connection * : Project Web Application

Method * : Resource; ReadResource

Data type returned: *ResourceDataSet*

**Input parameters**

resourceUid (Guid): ResourceUID

**Filter**

Table: ResourceCustomFields

Where the column: CODE_VALUE; Equals; Business unit Lookup Item UID

**Store results** *

Select column: Please select

**ResourceCustomFields**

CUSTOM_FIELD_UID: Resource Business Unit Custom Field UID

Figure 72 : Query Resource

The choice of extracting the **Custom Field UID** is arbitrary. In order to know the query was successful there needs to be some value returned. In this case it is stored in a variable that is tested by a **Run if** action to determine whether the Project Owner should be updated or not.

6. Drag a **Run If** action onto the design surface.
7. Drag the following actions into the **Run If** action boundary on the design surface:
    - Update project property
    - Publish project
    - Set variable

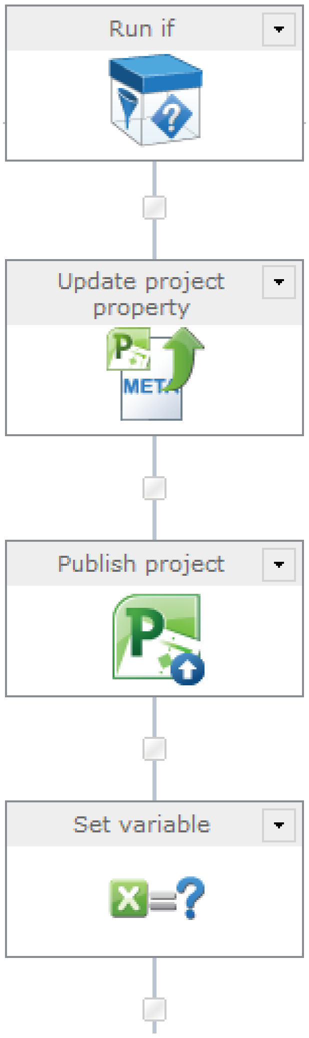
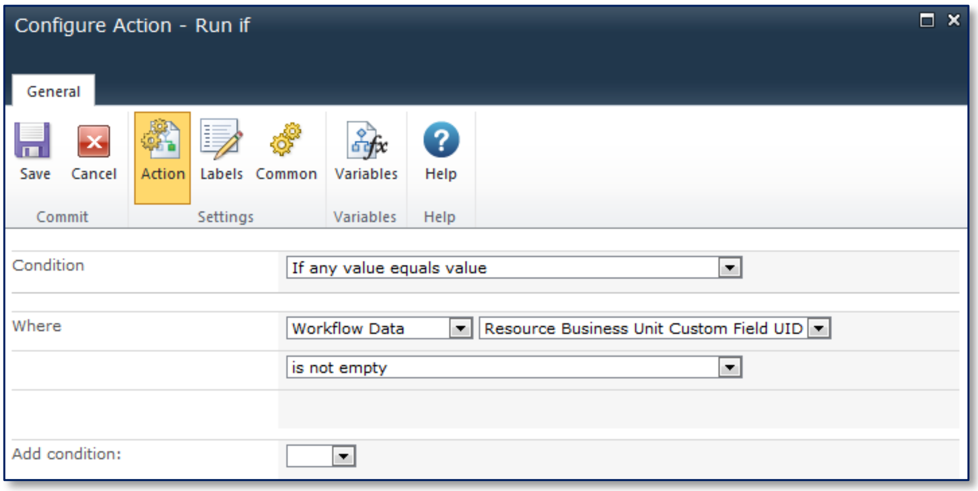


Figure 73 : Run If Action

8. Double-click on the **Run if** action and configure it as per below.
    a. In the **Condition** field, select **If any value equals value.**
    b. In the **Where** field, select **Workflow Data; Resource Business Unit Custom Field UID; is not empty.**



Condition: If any value equals value

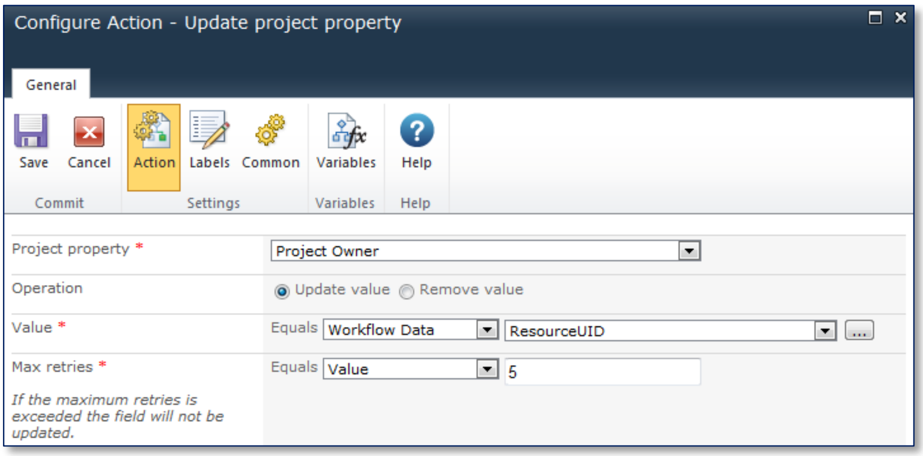
Where: Workflow Data | Resource Business Unit Custom Field UID

is not empty

Add condition:

Figure 74 : Run If Configuration

8. Configure the **Update project property** action to update the Owner of the Project to be the **ResourceUID** determined above.
    a. In the **Project property** field, select **Project Owner** and select **Update Value** as the operation.
    b. In the **Value** field, select **Workflow Data; ResourceUID.**
    c. In the **Max Retrieves** field, select **Value; 5.**



Project property *: Project Owner

Operation: ● Update value ○ Remove value

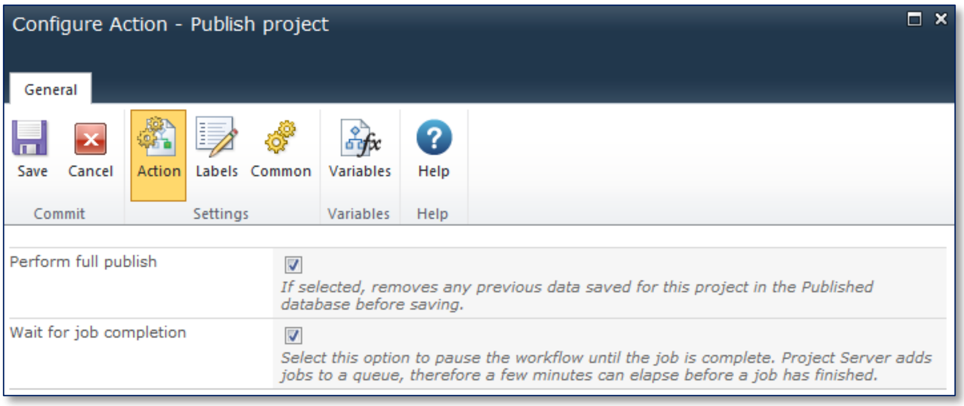
Value *: Equals Workflow Data | ResourceUID

Max retries *: Equals Value | 5

*If the maximum retries is exceeded the field will not be updated.*

Figure 75: Update Project Property

9. To ensure the data is published, a **Publish project** action can be used. Configure the action as below:



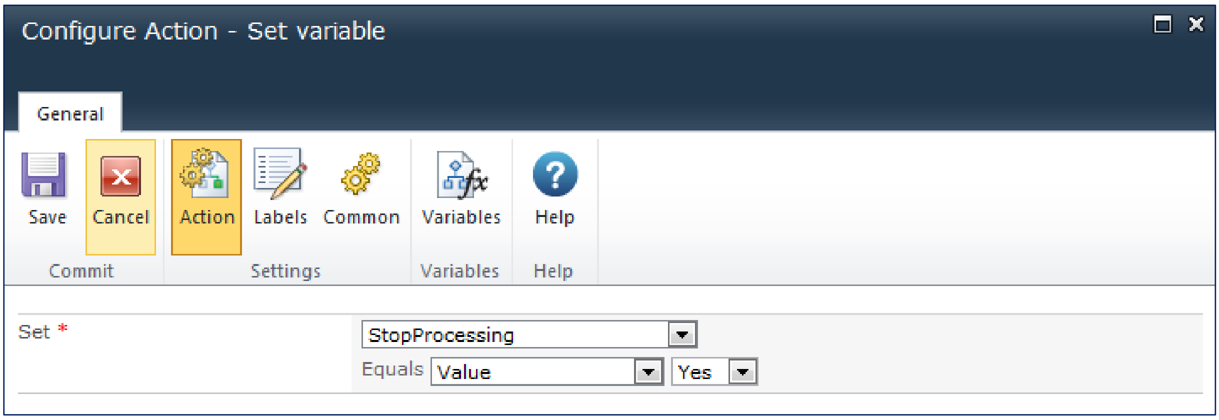
Perform full publish ☑
*If selected, removes any previous data saved for this project in the Published database before saving.*

Wait for job completion ☑
*Select this option to pause the workflow until the job is complete. Project Server adds jobs to a queue, therefore a few minutes can elapse before a job has finished.*

Figure 76 : Publish Project

10. Once a Project Manager for the chosen Business Unit is found, it is necessary to exit the **For Each loop** and close out the workflow, in this case by changing the **StopProcessing** variable to be true which will cause the loop to exit.

    To do so, double click on the Set variable action.
    From the set menu select **StopProcessing;equals Value Yes.**



Set * StopProcessing
Equals Value Yes

Figure 77 : StopProcessing variable

11. Finally, add a **Set Project Stage** action after the For Each loop and set the stage to be Execute.

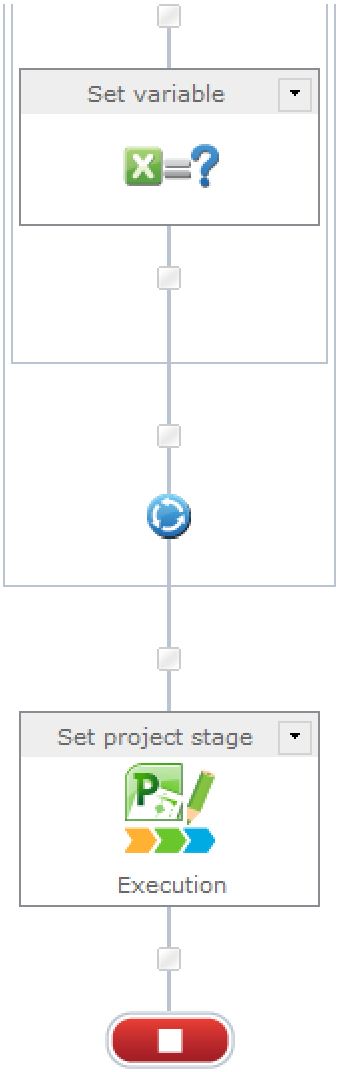


Figure 78 : Final Set Project Stage

The workflow is now ready to be published and associated to an Enterprise Project Type.

# Conclusion

This guide shows how Nintex Workflow for Project Server 2010 empowers Project and Portfolio Managers to design and implement Demand Management workflows without the need for complex and costly development cycles.

In Part Two of this series of product guides, we will be showing how Nintex Workflow for Project Server allows Project Managers to automate actions and alerts, and expedite decisions based on Project Server events.

Event driven alerts, tasks and actions help ensure greater governance across an organizations' project management practice. Nintex Workflow for Project Server workflows can be associated to asynchronous Project Server events to automatically start workflows when an event occurs in Project Server such as a project being published, or a new resource being created.

Nintex Workflow for Project Server 2010 helps you to execute your projects on time, within budget, aligned to your corporate strategies.

**Download a free 30 Day trial now at:** www.nintex.com/project.

# Appendix A - Setting up PWA Authentication in Central Administration

To centrally configure the PWA credentials that are used in the Query Project Server action:

1. Navigate to **Central Administration.**
2. In the **Nintex Workflow for Project Server 2010** section, click on **Manage Connections**.

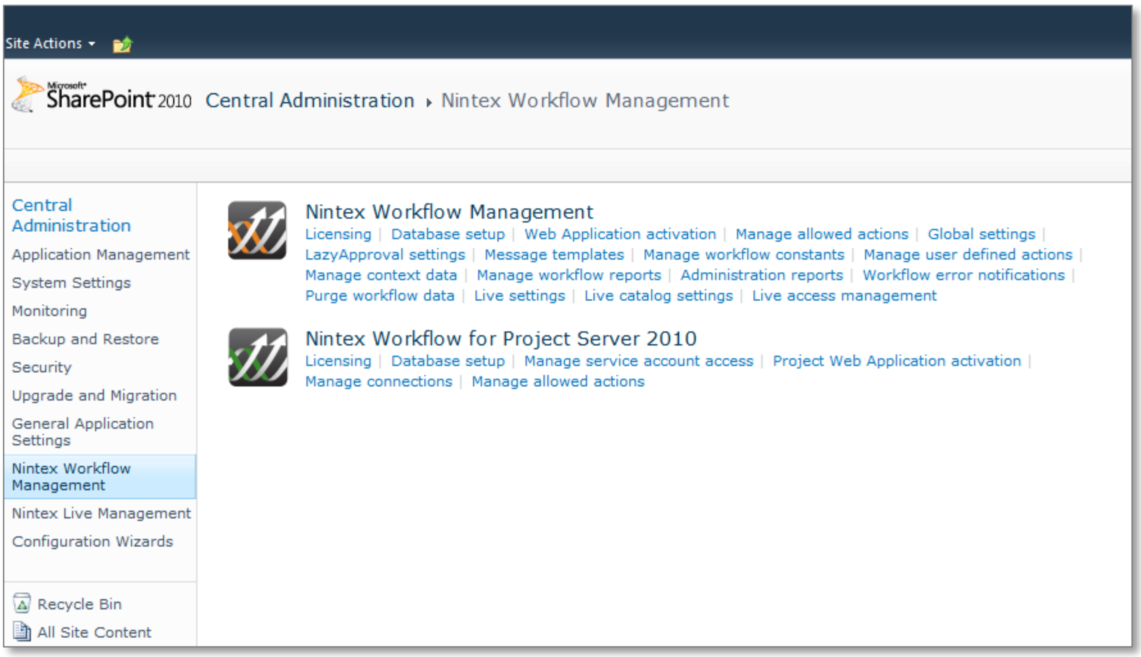

Site Actions
Microsoft SharePoint 2010
Central Administration ▸ Nintex Workflow Management
Central Administration
Application Management
System Settings
Monitoring
Backup and Restore
Security
Upgrade and Migration
General Application Settings
Nintex Workflow Management
Nintex Live Management
Configuration Wizards
Recycle Bin
All Site Content
Nintex Workflow Management
Licensing | Database setup | Web Application activation | Manage allowed actions | Global settings |
LazyApproval settings | Message templates | Manage workflow constants | Manage user defined actions |
Manage context data | Manage workflow reports | Administration reports | Workflow error notifications |
Purge workflow data | Live settings | Live catalog settings | Live access management
Nintex Workflow for Project Server 2010
Licensing | Database setup | Manage service account access | Project Web Application activation |
Manage connections | Manage allowed actions

Figure 79: Nintex Workflow Management in Central Administration

Nintex Workflow for Project Server 2010 allows multiple connections to be configured; this is especially useful if you have more than one Project Server instance operating within your Project Server farm.

3. To configure a connection, click on **New**.

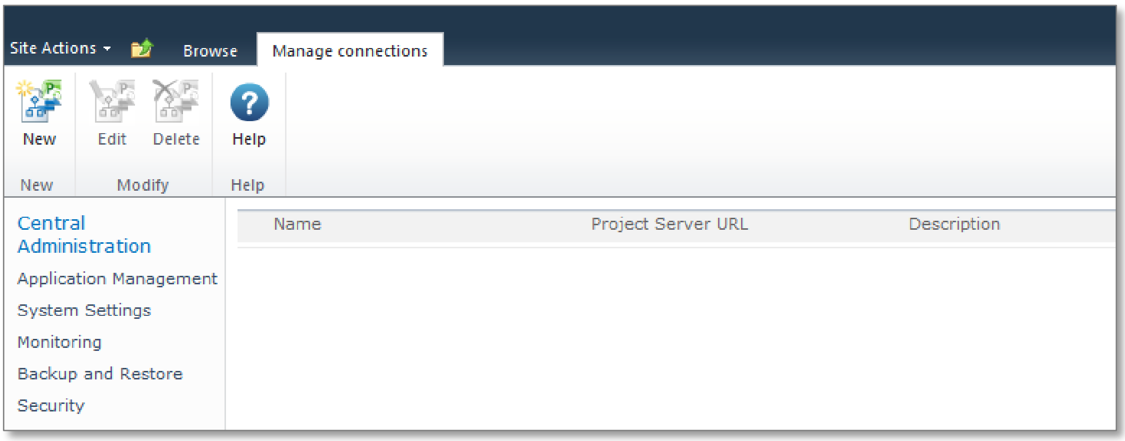
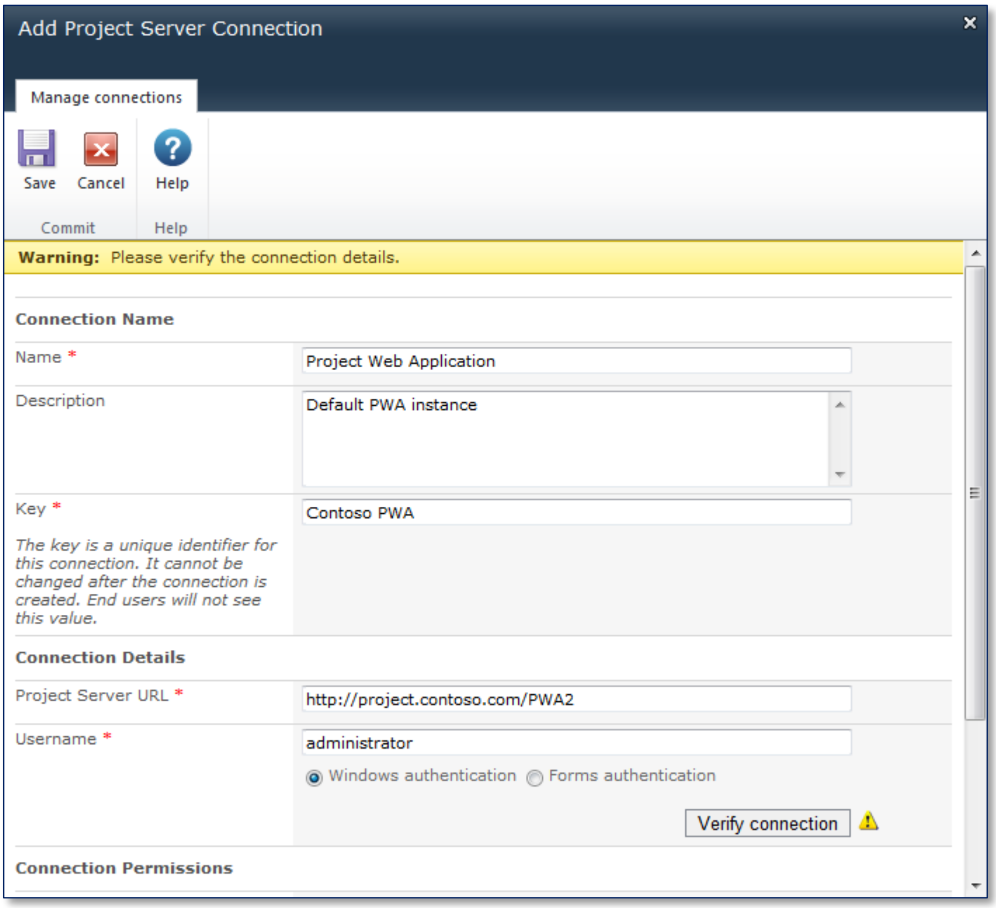


Figure 80: Manage Connections

The Connection Configuration screen will be displayed. Enter the relevant settings, taking care to ensure the connection is verified before saving.



**Warning:** Please verify the connection details.

**Connection Name**

Name * : Project Web Application

Description: Default PWA instance

Key * : Contoso PWA

*The key is a unique identifier for this connection. It cannot be changed after the connection is created. End users will not see this value.*

**Connection Details**

Project Server URL * : http://project.contoso.com/PWA2

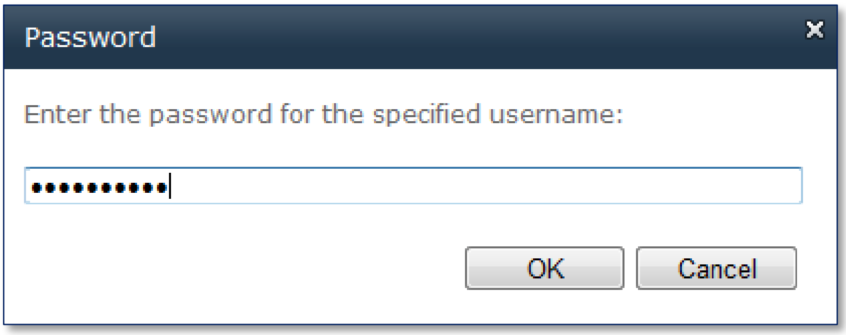
Username * : administrator

● Windows authentication ○ Forms authentication

**Connection Permissions**

Figure 81: Add Project Server Connection

4. Only once the connection is verified will the user be prompted to supply the password to use for the connection.

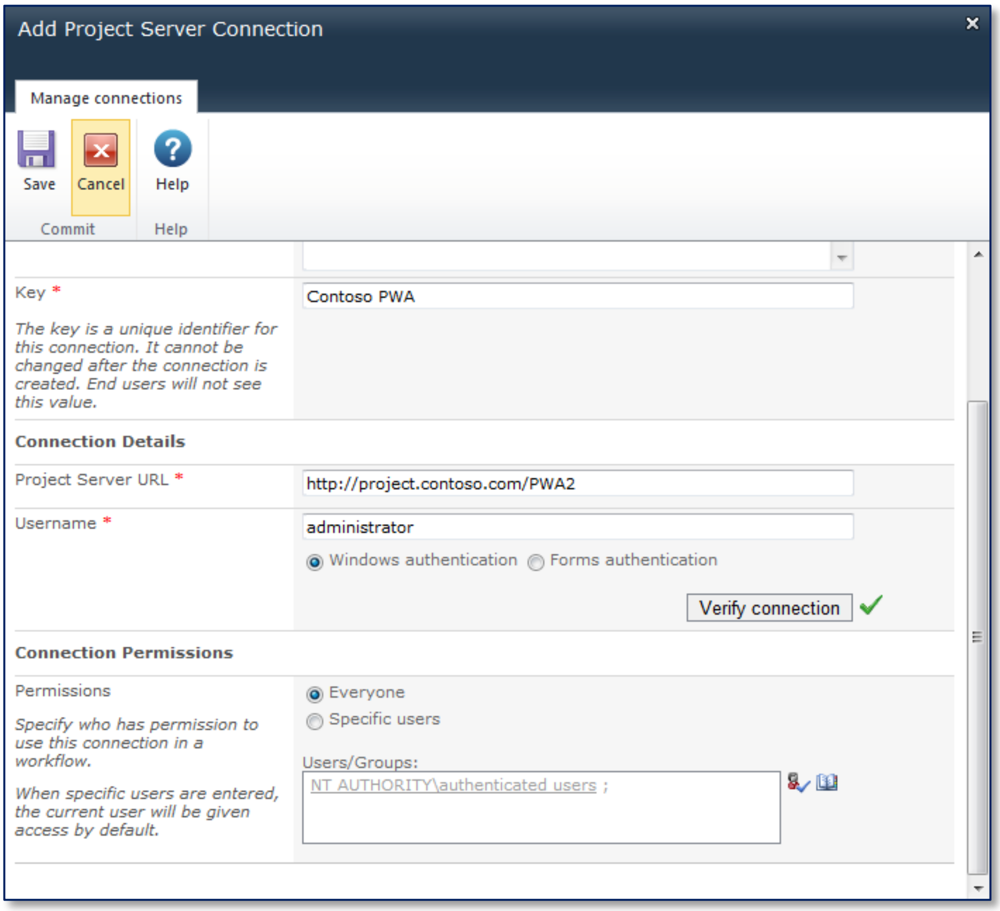


Enter the password for the specified username:

•••••••••••

Figure 82: Enter Connection Password

5. Finally, the connection requires the administrator to define which users have access to use the connection. This allows the administrator to restrict access to the action for users of different instances.



Key *

Contoso PWA

*The key is a unique identifier for this connection. It cannot be changed after the connection is created. End users will not see this value.*

**Connection Details**

Project Server URL * http://project.contoso.com/PWA2

Username * administrator

● Windows authentication ○ Forms authentication

✓

**Connection Permissions**

Permissions

● Everyone
○ Specific users

*Specify who has permission to use this connection in a workflow.*

*When specific users are entered, the current user will be given access by default.*

Users/Groups: NT AUTHORITY\authenticated users ;

Figure 83: Connection Permissions

6. Once completed, click on **Save**.

The configured connection will then be available in the Manage Connection screen and for selection in the **Query project server** actions.

# Appendix B – References

| Item | Location |
|---|---|
| Microsoft Playbooks Tool – Part of the Microsoft Project Server 2010 Resource Kit | Overview of the Playbooks tool (http://technet.microsoft.com/en-us/library/gg128952.aspx) |
| | Project Resource Kit (x64) & (x86) - http://www.microsoft.com/en-us/download/details.aspx?id=22810 |
| DM Demand Management Solution starter | http://archive.msdn.microsoft.com/P2010SolutionStarter |