
Hitchhiker's Guide to Demand Management 2013

Project 2013 Case Study

Prepared for

Contoso

February 2013

Prepared by

Steven Haden

Published by

[Project Server Content Publishing](#)



The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

The descriptions of other companies' products in this document, if any, are provided only as a convenience to you. Any such references should not be considered an endorsement or support by Microsoft. Microsoft cannot guarantee their accuracy, and the products may change over time. Also, the descriptions are intended as brief highlights to aid understanding, rather than as thorough coverage. For authoritative descriptions of these products, please consult their respective manufacturers.

© 2013 Microsoft Corporation. All rights reserved.

Microsoft, SharePoint, Visio, Visual Studio, and Windows are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Table of Contents

1	INTRODUCTION.....	9
1.1	Purpose	9
1.2	Audience	9
1.2.1	Reading Guide	9
1.3	General steps	10
1.4	What's New with Project Server 2013 workflow	10
1.4.1	New Tools available: SharePoint Designer 2013 and Visio 2013	10
1.4.2	No more Code	11
2	VISION: CASE STUDY FOR CONTOSO	12
2.1	Introduction	12
2.2	Using Visio 2013.....	13
3	PLAN/SPECIFY: THE DEMAND MANAGEMENT PROCESS LIFECYLE.....	17
3.1	Introduction	17
3.2	Overview	17
3.3	Define the process	18
3.3.1	Phases	18
3.3.2	Stages	19
3.3.3	Activities.....	21
3.4	The Contoso Sample Project	22
3.5	Objects to configure and use in Project 2013.....	23
4	BUILD/CREATE: WORKFLOW CONFIGURATION.....	25

4.1	Enterprise Project Templates.....	25
4.2	Phase: A Collection of Steps in a Project Life Cycle	25
4.3	Stage: A Step in a Project Life Cycle.....	25
4.4	Project Detail Pages (PDPs) in Stages.....	25
4.5	Configuration tasks on Project Server 2013	26
4.5.1	Initial Setup (On-premises)	26
4.5.2	Workflow configuration	26
4.6	Configuration Best practices.....	27
4.6.1	Naming conventions to use to distinguish specific workflow objects	27
4.6.2	List of custom fields used in our example.....	29
4.7	How to create workflow objects in Project Online.....	29
4.7.1	Lookup tables and custom fields	30
4.7.2	Project Detail Pages Creation.....	36
4.7.3	Workflow Phases Creation.....	42
4.7.4	Workflow Stages Creation	44
4.7.5	Enterprise Project Type creation	50
5	WORKFLOW ORCHESTRATION: THE BIG PICTURE.....	52
5.1	Creating Project 2013 workflows.....	52
5.2	Workflow Key concepts	53
5.3	Workflow Creation, Administration, and Usage Process.....	54
5.4	Relations between Project Server Objects and Workflow orchestration.....	54
5.5	Decision tree between SharePoint Designer and Visual Studio	56
5.5.1	Comparing SharePoint Designer workflows versus Visual Studio Workflow	57

5.6	Using SharePoint Designer 2013 to import Visio Diagrams.....	60
5.7	Using SharePoint Designer 2013.....	62
5.7.1	Using a new site workflow	62
5.7.2	Using the Visio import.....	64
5.8	Using Visual Studio 2012 to create Project Workflows	71
5.9	Deploy the workflow (using SharePoint Designer)	79
5.10	Understanding the Skip to Stage steps	80
5.11	How to build your own custom actions to use in SharePoint Designer.....	86
5.11.1	Overview of a custom action	87
5.11.2	Writing custom activities in Visual Studio 2012.....	87
5.11.3	Updating and deleting custom actions	87
6	<i>USING SHAREPOINT LISTS TO CAPTURE DEMAND.....</i>	88
6.1	Capture the demands in SharePoint List.....	88
6.2	Create an Enterprise Project from a SharePoint List	88
7	<i>DEPLOY: DEPLOYMENT FROM DEV TO QA to PRODUCTION</i>	92
7.1	Copying a Workflow	92
8	<i>DEBUG/MONITOR: TROUBLESHOOTING WORKFLOWS</i>	93
8.1	Troubleshooting a Workflow	94
8.1.1	Check the workflow status page	95
8.1.2	View the ULS logs (only available on-premises)	96
9	<i>REFERENCES.....</i>	97
9.1	General references.....	97
9.2	References in this document	98

Table of Figures

Figure 1: List of users	10
Figure 2: Contoso Sample Project Process.....	12
Figure 3: SharePoint 2013 Template in Visio 2013	14
Figure 4: The initial Stage shape in Visio 2013	14
Figure 5: List of Actions (Components and Conditions) in Visio 2013	15
Figure 6: Stage outline process view in Visio 2013	16
Figure 7: Stage View in Vision 2013	16
Figure 8: Full Contoso example in the Stage Outline View	17
Figure 9: Demand Management phases and stages	18
Figure 10 : List of phases.....	18
Figure 11: List of stages.....	19
Figure 12: List of fields for Idea Collection.....	20
Figure 13: List of fields for Lead Approval.....	20
Figure 14: List of fields for Cancelled	21
Figure 15: List of fields for Execution	21
Figure 16: List of fields for Post Mortem	21
Figure 17: List of activities.....	22
Figure 18: Contoso Sample Process (detailed).....	23
Figure 19: List of objects to configure/use in Project Server 2013	24
Figure 20: Hierarchy of Project Server objects	26
Figure 21: IT workflow phases	27
Figure 22: IT workflow stages	28
Figure 23: Contoso Sample Project example phases	28
Figure 24: Contoso Sample Project example stages	28
Figure 25: Contoso list of custom fields	29
Figure 26: Project Web App Online tenant	29
Figure 27: List of CF and LT for Contoso sample	31
Figure 28: Steps to create CF and LT	36
Figure 29: Steps to create PDPs	42
Figure 30: Steps to create phases	43
Figure 31: Steps to create stages	49
Figure 32: Steps to create an EPT.....	51
Figure 33: High level steps to create a Project Server 2013 workflow	52
Figure 34: SharePoint 2013 Workflow Architecture	53
Figure 35: Different actors for managing a workflow	54
Figure 36: Relation between Project Server and Visual Studio/SPD objects	55
Figure 37: Comparing SharePoint Designer and Visual Studio for implementing Workflows	57

Figure 38: Features and Requirements comparison between SharePoint Designer and Visual Studio	60
Figure 39: Using SharePoint Designer 2013	60
Figure 40: Importing the workflow from Visio 2013	61
Figure 41: Creating a workflow from a Visio 2013 import	61
Figure 42: Result of Visio 2013 import	62
Figure 43: Creating a site workflow in SharePoint Designer 2013	62
Figure 44: SharePoint Site Workflow Dialog box	63
Figure 45: New workflow using Text-Based Designer	63
Figure 46: Using the PWA defined stages from SharePoint Designer 2013	63
Figure 47: Default Stage Outline	64
Figure 48: SharePoint Designer Properties	64
Figure 49: Selecting the right PWA stage from SharePoint Designer	65
Figure 50: Workflow in SharePoint Designer after remapping the correct PWA stage	65
Figure 51: Check for errors in SharePoint Designer 2013	66
Figure 52: No errors in SharePoint Designer	66
Figure 53: Generate Workflow outline view	66
Figure 54: Result of Workflow outline	67
Figure 55: Switching to Text-Based Designer	67
Figure 56: Initial Workflow in Text-Based Designer	68
Figure 57: Testing if a Funding is required from SharePoint designer 2013	69
Figure 58: Using a sending email Action	69
Figure 59: Generated email when the project is approved	70
Figure 60: Workflow in Text-Based Designer after updates	71
Figure 61: Steps to create a Project Server 2013 workflow with Visual Studio 2012	78
Figure 62: Publishing the workflow to PWA from SharePoint Designer 2013	79
Figure 63: Associating the workflow to an EPT	80
Figure 64: Example of skip to stage workflow	81
Figure 65: Result of skip to stage workflow	82
Figure 66: Restarting the workflow and skipping to stage Test Execute Stage 2	82
Figure 67: Result of workflow after skipping to Stage Test Execute Stage 2	83
Figure 68: Using the "Skip until the current workflow stage"	83
Figure 69: Result after using "Skip until the current workflow stage"	84
Figure 70: Workflow example with skipping in transition	85
Figure 71: Result of running with skipping in transition	86
Figure 72: Result of running to any stage other than Create 1	86
Figure 73: Create Projects from a SharePoint list	89
Figure 74: Create Projects dialog box to map list columns to Project Web App enterprise custom fields	89
Figure 75: Project Server settings to define list mapping	90
Figure 76: Field mappings column name to PWA Fields	90
Figure 77: Dialog box after creation of projects	91
Figure 78: Continue the standard workflow process	91
Figure 79: Remapping the PWA2 objects	93

Figure 80: Logging a message to the workflow history list	94
Figure 81: Formatting your message to display in Log History	94
Figure 82: Steps to troubleshoot a workflow.....	96

1 INTRODUCTION

1.1 Purpose

This document contains the information necessary to get started on implementing a full Demand Management solution with Microsoft Project Server 2013.

It assumes that no previous configuration is defined in a Project Web App instance.

This article applies both to a Project Server 2013 on-premises deployment and also to the newly available Office 365 Project Online environment.

This document is an updated version of the white paper that was published for Project Server 2010:

[Hitchhiker's Guide to Demand Management](http://technet.microsoft.com/en-us/library/ff973112(v=office.14).aspx) ([http://technet.microsoft.com/en-us/library/ff973112\(v=office.14\).aspx](http://technet.microsoft.com/en-us/library/ff973112(v=office.14).aspx)).

1.2 Audience

This document is designed for the following types of users:

- IT Professionals
- Developers
- Project Management Office (PMO) or PWA administrators
- Business analysts
- Business users

1.2.1 Reading Guide

Figure 1 shows the chapters that apply for each type of user:

Chapter /User	IT Professionals	Developers	PMO or PWA admin	Business Analyst/Process consultant	Business User
Vision			X	X	X
Plan/Specify		X	X	X	X
Build/Create		X	X	X	
Configuration					

Build/Create		X			X
Orchestration					
Deploy	X	X	X		X
Debug/Monitor	X				
Use					X

Figure 1: List of users

1.3 General steps

There are four general steps to perform to create your workflow in Project Server 2013 or Office 365 Project Online that are detailed in this document:

1. Plan/Vision: Using Visio 2013
2. Workflow Configuration: Create objects in Project Server
3. Workflow Orchestration: Create workflow in SharePoint Designer 2013
4. Deploy the Workflow

1.4 What's New with Project Server 2013 workflow

Project Server 2013 workflows are built on the SharePoint 2013 workflow platform, which use version 4 of Windows Workflow Foundation (WF4). Unlike in previous versions, declarative workflows for Project Server 2013 can be created by using SharePoint Designer 2013 and are accessible for both on-premises and online use. Project Server workflows use the SharePoint workflow security model with OAuth, and can be installed on a Project Web App site.

It is no longer necessary to implement workflows by using Visual Studio.

Project Server 2013 workflows are implemented in the new SharePoint 2013 architecture. For more information see [What's new in workflow in SharePoint Server 2013](http://technet.microsoft.com/en-us/library/jj219638(office.15)) ([http://technet.microsoft.com/en-us/library/jj219638\(office.15\)](http://technet.microsoft.com/en-us/library/jj219638(office.15)))

Declarative workflows can run either in Project Server 2013 on-premises or in Project Online.

You can also directly capture the demand by using a SharePoint list, as described in chapter 6 ([Using SharePoint Lists to Capture Demand](#)).

1.4.1 New Tools available: SharePoint Designer 2013 and Visio 2013

Visio 2013 can be used to design workflows by a business analyst or a business user.

By using Visio 2013, you can visually create a Project workflow, export the workflow to SharePoint Designer 2013, and then publish that workflow to a Project Web App site.

After a workflow is created in Visio 2013, it must be exported to SharePoint Designer 2013. Then using SharePoint Designer 2013, a PWA site administrator or IT professional adds parameters to the workflow by using either the workflow text editor or the new Visual Workflow Designer, which is a Visio 2013 ActiveX control that is hosted in SharePoint Designer 2013. After the workflow is completed, it can be published to the PWA site.

This is ideal for business analysts and process consultants who are already familiar with flowcharts in Visio, because it allows them to design a workflow that represents business logic. The person who designs the workflow can focus solely on the business intelligence (BI) needs of the workflow without needing to be an expert in declarative workflows.

SharePoint Designer 2013 provides a visual workflow designer. The previous version, SharePoint Designer 2010, offered integration with Visio 2010, so that you could design workflows visually in Visio and then import them into SharePoint Designer. Now, you can visually design your workflows by dragging, dropping, and connecting shapes directly within SharePoint Designer 2013. You can set all the properties of a workflow action in the new visual workflow designer—just select the shape and then click the action tag. Also, if you click **Properties** on the action tag, the property grids for conditions and actions appear in the visual designer, just as they do in the declarative workflow designer.

From SharePoint Designer 2013, a connection can be made to a Project Web App instance to retrieve the relevant objects that are configured in Project Web App.

The workflow can be published from SharePoint Designer 2013 to the Project Web App instance.

For more information and instructions on how to create a sample Project Server workflow by using SharePoint Designer 2013, see [Getting started developing Project Server 2013 workflows](http://msdn.microsoft.com/en-us/library/ee767694.aspx) (<http://msdn.microsoft.com/en-us/library/ee767694.aspx>).

1.4.2 No more Code

Using SharePoint Designer 2013, a Project Server Workflow can be implemented by using only SharePoint Designer. It is no longer necessary to use Visual Studio and a .NET Framework language.

Visual Studio is no longer required to develop Project Server workflows that use custom fields, stages, phases, and enterprise project types that can be managed in Project Web App. Although you can use Visual Studio 2012 to develop workflows, they are often easier and quicker to create by using SharePoint Designer. Visual Studio can be used for workflows that require access to the client-side object model (CSOM) for Project Server, or other external APIs.

You build a declarative workflow by adding workflow stages, actions, conditions, and other elements in a design tool, which can be either SharePoint Designer 2013 or Visual Studio 2012. The design tool saves the workflow as XAML code, which is interpreted at runtime. By using Visual Studio 2012, you can also build custom actions and forms for additional control, and save workflow templates for reuse with multiple Project Web App instances. SharePoint Designer 2013 can consume custom actions that are created in Visual Studio 2012.

2 VISION: CASE STUDY FOR CONTOSO

2.1 Introduction

In this chapter we define the demand management process to be used by Contoso.

It is not the goal of this document to describe how to define the demand management process; this subject is covered by the document [Demand Management in Project Server 2010](http://go.microsoft.com/fwlink/?LinkId=191854) (<http://go.microsoft.com/fwlink/?LinkId=191854>).

We use the simple example in Figure 2 to model the project process from the idea collection to its execution and a post-mortem:

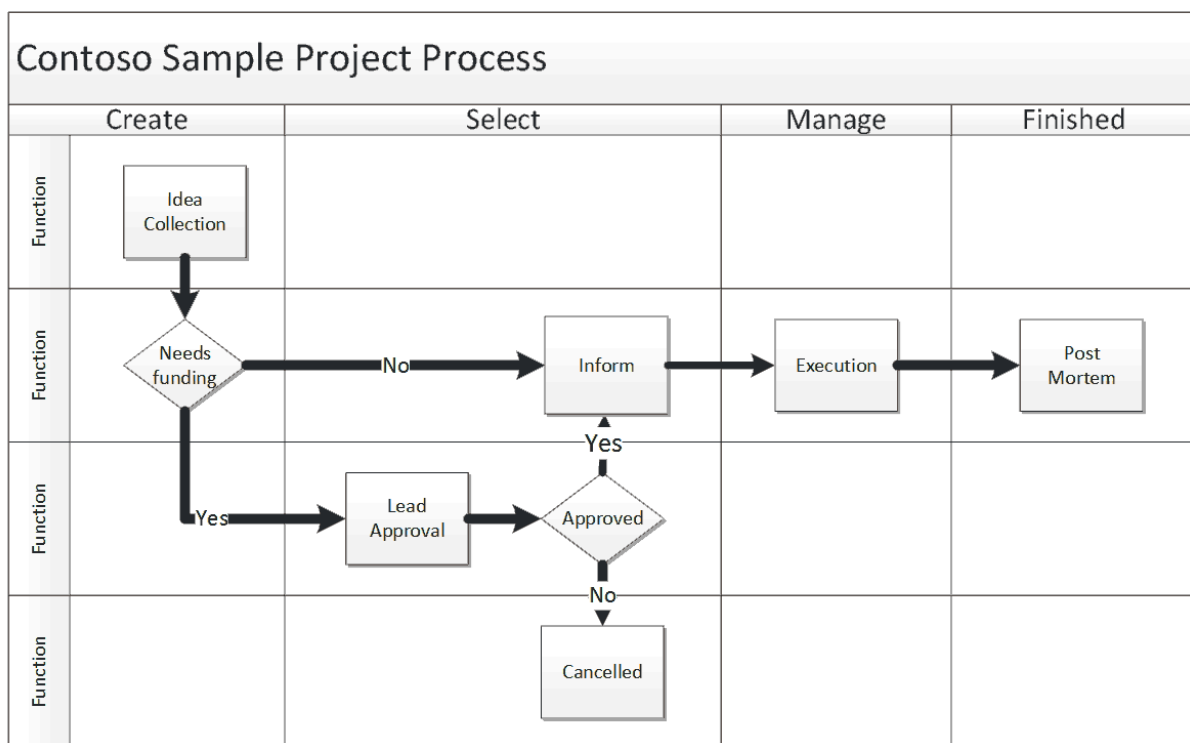


Figure 2: Contoso Sample Project Process

The Demand Management process used by Contoso starts in the Create phase by collecting information about the new idea: name, description, choice for required funding, proposed cost and project manager.

If the demand requires funding, we enter the Select phase where it can be approved by a Team Lead. After the demand is approved or if the demand requires no funding we inform the project manager about the new assignment. The project owner is notified by an email that she is in charge of a new project. If the demand is not approved, the project is cancelled.

If the demand is approved, or if no funding is required, we enter the Manage phase.

The project manager then runs and schedules the project until it is completed.

When the project is completed, we enter the Finish phase, where information is captured about the project execution in a post-mortem stage.

2.2 Using Visio 2013

The Demand Management specification can be done by using Visio 2013 and the new SharePoint 2013 Workflow template. In Visio 2013, a Project Server workflow must be a site workflow, and must use the **SharePoint 2013 Workflow – Project Server** platform type.

All of the master shapes in the SharePoint 2013 Workflow stencils correspond to actions, conditions, and other logical constructs within a SharePoint 2013 workflow. To build a workflow, you can drag shapes onto the drawing canvas, just like any other flowchart in Visio. After you finish building the workflow in Visio 2013, you must save the workflow before SharePoint Designer 2013 can import it.

To open the SharePoint 2013 Workflow template in Visio 2013, do the following:

1. Open Visio 2013.
2. Choose **New**.
3. Under **Template Categories**, choose **Flowchart**.
4. Under **Choose a Template**, choose SharePoint 2013 **Workflow** and then choose **Create**.

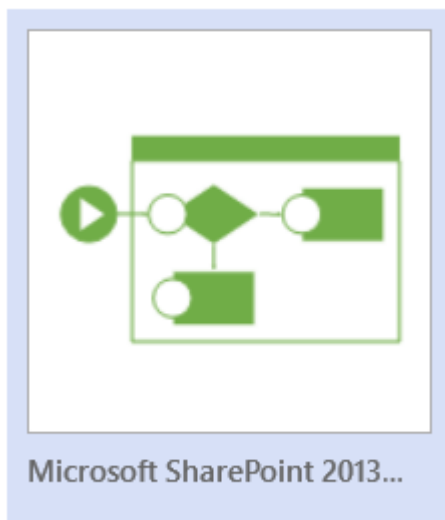


Figure 3: SharePoint 2013 Template in Visio 2013

The template opens and the drawing canvas is prepopulated with a Start shape and a Stage shape. The Stage shape contains an Enter and an Exit shape, joined by a single connector.

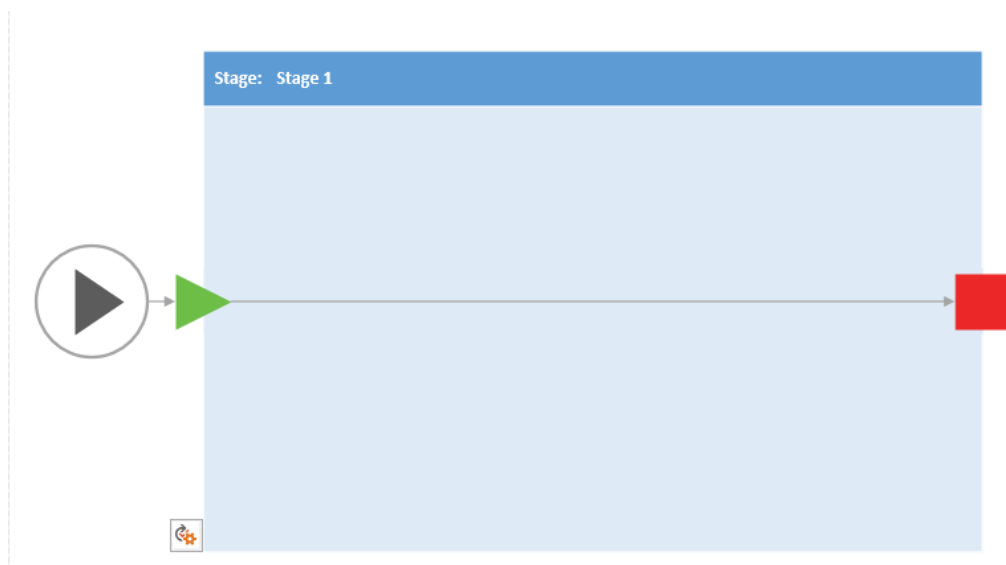


Figure 4: The initial Stage shape in Visio 2013

By using this template you can define the following in your workflow:

Project-specific Actions

- Create a project from current item
- Set project field
- Set project stage status
- Wait for project event

Project condition

- Skip project stage

Actions - SharePoint 2013 Workflow

Conditions - SharePoint 2013 Workflow

Components - SharePoint 2013 Workflow

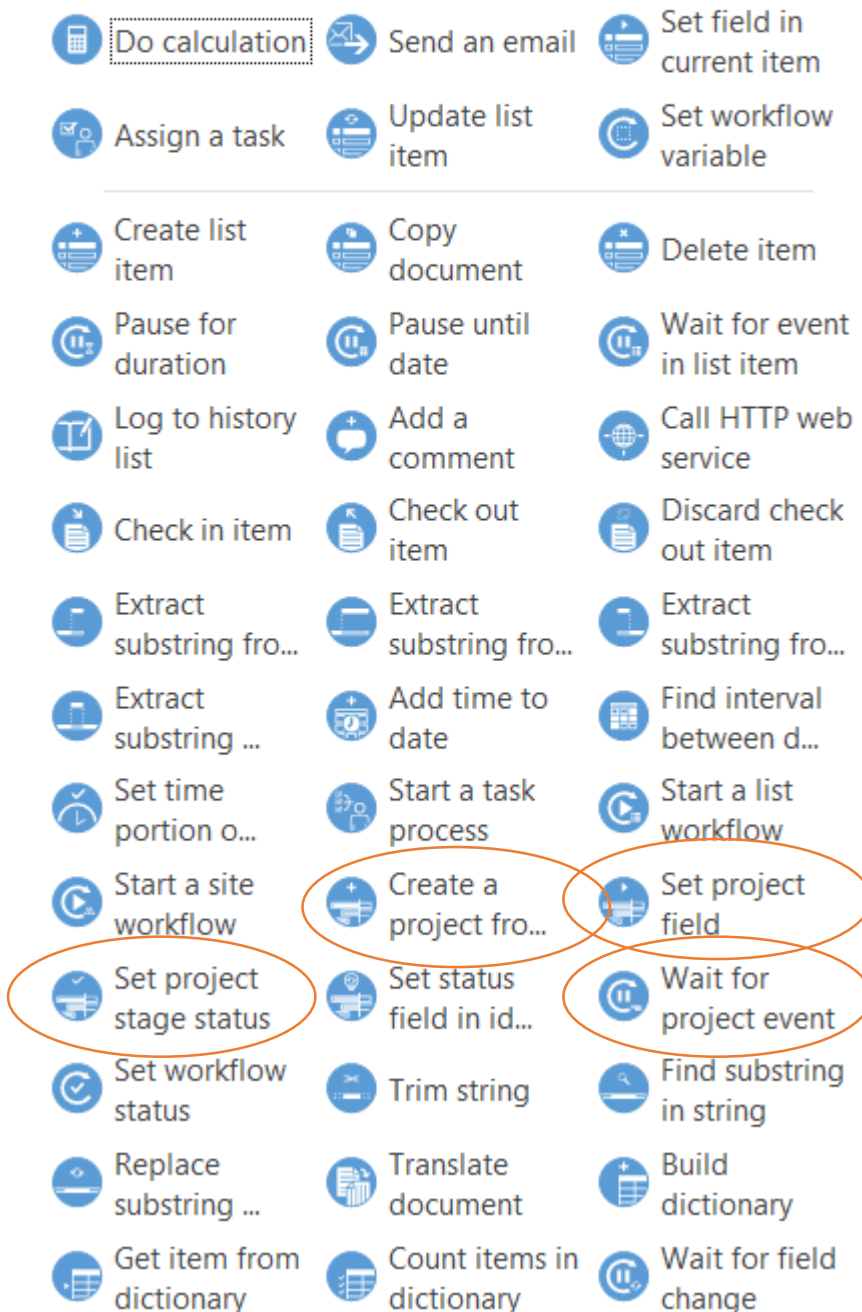


Figure 5: List of Actions (Components and Conditions) in Visio 2013

In Visio 2013, there is no link to the Project Server 2013 web app instance.

The business analyst can work at a high level using the new Stage Outline process view:

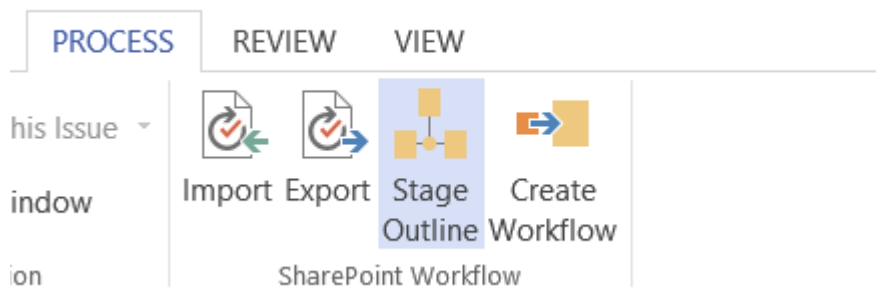


Figure 6: Stage outline process view in Visio 2013

This view enables you to work at a high level in the business process workflow, without going into the implementation details of the different workflow actions.

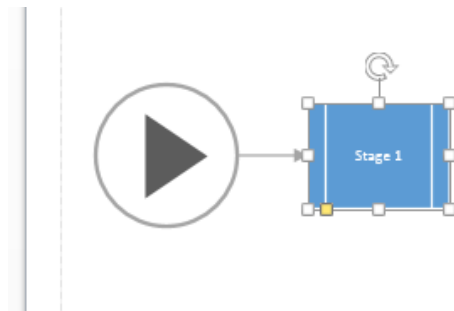


Figure 7: Stage View in Visio 2013

So if we implement the example presented in chapter 2.1, we would have the following workflow defined in Visio:

- The six stages with two conditions:
 - Idea collection
 - Lead Approval
 - Inform
 - Cancelled
 - Execution
 - Post-mortem

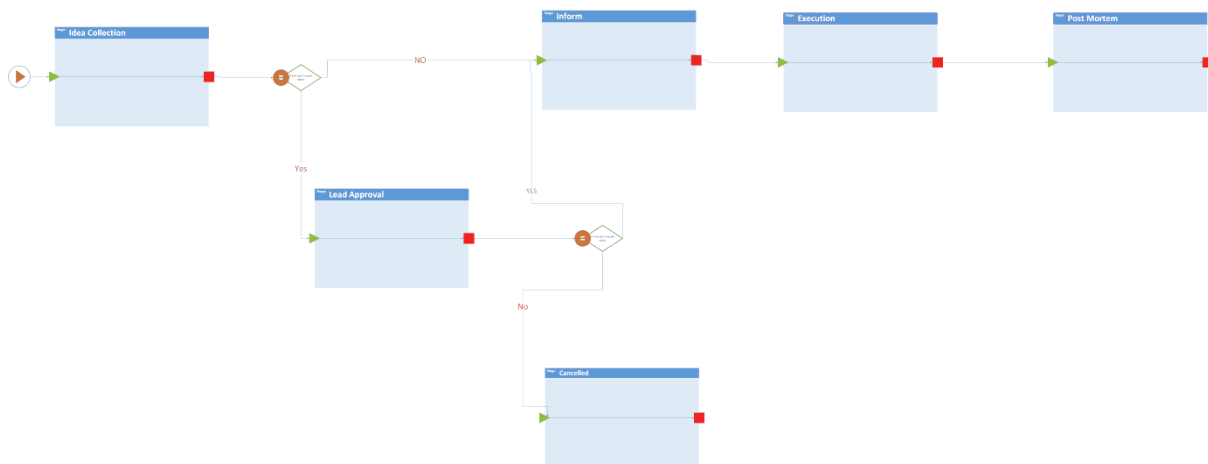


Figure 8: Full Contoso example in the Stage Outline View

The Visio file can be saved in the normal Visio 2013 file format (a .vsdx Visio drawing) which can be imported in SharePoint Designer 2013.

More information can be found in the following MSDN article: Workflow development with SharePoint Designer or Visio 2013 ([http://msdn.microsoft.com/en-us/library/jj163272?v=\(office.15\).aspx](http://msdn.microsoft.com/en-us/library/jj163272?v=(office.15).aspx))

3 PLAN/SPECIFY: THE DEMAND MANAGEMENT PROCESS LIFECYCLE

3.1 Introduction

The demand management process captures all work proposals in one place, guides the proposals through a multistage governance process, helps users make decisions about which proposals to approve, and tracks progress on project execution until the work is completed. A key component within demand management is the workflow governance model implemented in Project Server.

A governance workflow includes definitions of the life cycle stages through which the project progresses, such as proposal creation and initial approval. The workflow defines what information is required or locked in each stage. For example, a workflow can lock budget cost after the project is approved. A workflow can include necessary manual approval or notification steps and add business logic to update other LOB systems. A workflow can also update an enterprise resource planning (ERP) system when the proposal budget is approved.

3.2 Overview

In project portfolio management, a project life cycle is a long-running process that spans governance phases. Typical demand management phases are: create, select, plan, and manage. The planning and management phases are accomplished by the more familiar project management processes by using Project Professional and Project Web App. Workflows model the governance processes and provide a structured way for projects to proceed through the phases. Workflows, along with other proposal data in project detail pages (PDPs), are captured and integrated within the demand management feature set, providing a rich and dynamic platform on which customers and partners can build custom solutions.

Figure 9 shows the four standard phases of demand management and how they fit together. Within each phase are stages such as *Propose idea* and *Initial review*. Each stage can have one or more associated PDPs in Project Web App. The entire collection of stages represents a single workflow that can be linked to an enterprise project template (EPT).

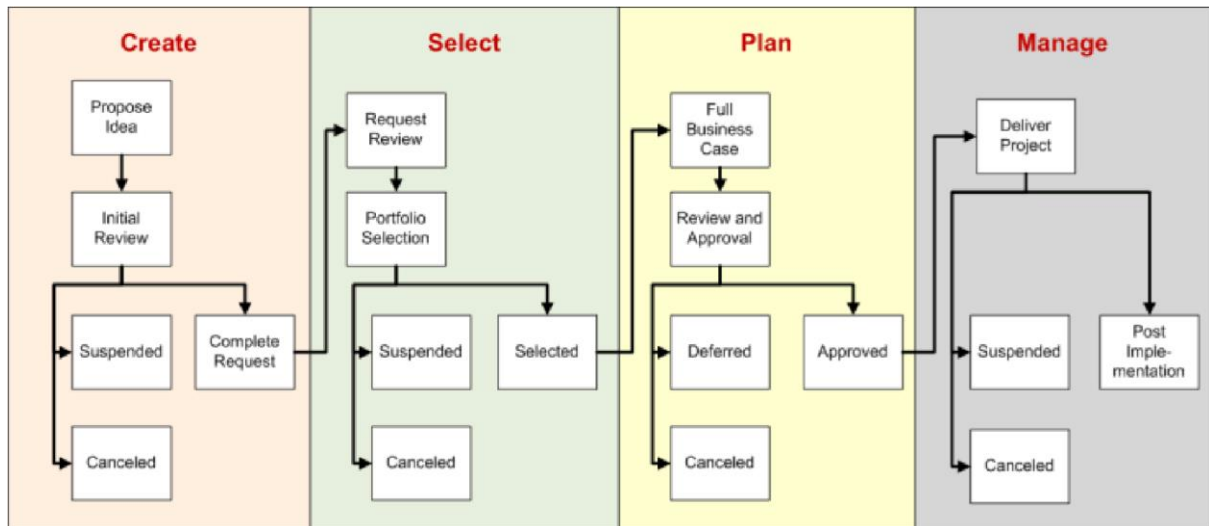


Figure 9: Demand Management phases and stages

3.3 Define the process

We first define the process to be used in terms of phases, stages, and activities.

We recommend that you first use a business modeling tool such as Visio to outline the project processes. The previous chapter shows an example.

In the Vision chapter, we have defined the Process for our simple project.

We will now refine this vision.

3.3.1 Phases

In our example we have identified the following phases:

Phase name	Phase Prefix
Create	C
Select	S
Manage	M
Finished	F

Figure 10 : List of phases

We use a simple prefix, of one or two letters for each phase, which enables us to use the prefix when referring to the phase from another context.

3.3.2 Stages

For each stage, we define the inputs to the stage and the outputs that result when the stage is complete.

The inputs and outputs define the list of fields that are displayed in the Project Detail Page (PDP). At the same time (if necessary) we can list the resulting outputs from this stage, such as computed fields. The following steps list the general procedure:

1. Define the input and output of each stage
2. Define the required information to capture (fields) in forms
3. Define the state of the fields in each stage (Required, R/W, Read-only)

A stage controls the behavior of:

- Visible PDPs
- Read-Only/Required Custom Fields
- Associated Phases

In our example we define the following stages:

Stage name	Stage Prefix	Inputs	Outputs
Idea Collection	IC	Name, Business Reason, Needs Funding, Proposed Cost, Project Manager/Owner	
Lead Approval	LA	Decision	
Cancelled	C	N/A	
Execution	E	Planning/Progress	
Post Mortem	PM	Lessons Learned, On Budget, Reasons	

Figure 11: List of stages

For each stage in our example, we detail the different fields.

Idea Collection Input	Required	State	Output	Format
Name	Yes	R/W		String 255 characters
Description	Yes	R/W		String Multiple lines
Needs Funding	Yes	R/W		Yes/No choice
Proposal Cost	Yes	R/W		Decimal/Cost
Project Manager/Owner	Yes	R/W		String 255 characters

Figure 12: List of fields for Idea Collection

Lead Approval Input	Required	State	Output	Format
Name	N/A	R		String 255 characters
Description	N/A	R		String Multiple lines
Project Manager/Owner	N/A	R		String 255 characters
Decision	Yes	R/W		Yes/No

Figure 13: List of fields for Lead Approval

Cancelled Input	Required	State	Output	Format
N/A				

Figure 14: List of fields for Cancelled

Execution Input	Required	State	Output	Format
N/A				

Figure 15: List of fields for Execution

Post Mortem Input	Required	State	Output	Format
Lessons Learned	Yes	R/W		String Multiple line
On Budget	Yes	R/W	Yes/No	Choice
Reasons		R/W		String Multiple line

Figure 16: List of fields for Post Mortem

3.3.3 Activities

For each stage we define whether specific activities are needed.

Some examples of possible activities are:

- Define whether an approval is needed
- Define whether e-mail messages are sent
- Define which information on the workflow progress is communicated to the users

In our example here are the lists of activities we have identified:

Stage name	Activities List	Progress information
------------	-----------------	----------------------

Idea Collection	Needs funding Approval
Lead Approval	Approval
Inform	E-mail to inform Project Manager/Owner
Cancelled	Cancelled
Execution	
Post Mortem	

Figure 17: List of activities

3.4 The Contoso Sample Project

Figure 18 shows the Visio diagram of the Contoso sample project workflow. This Visio diagram is only used for documentation purposes and not to be used as a SharePoint Designer 2013 input.

This diagram captures the different objects that are present in our workflow solution:

- Phases
- Stages
- Activities
- PDPs visible for each activity and the state: Read-only or R/W.

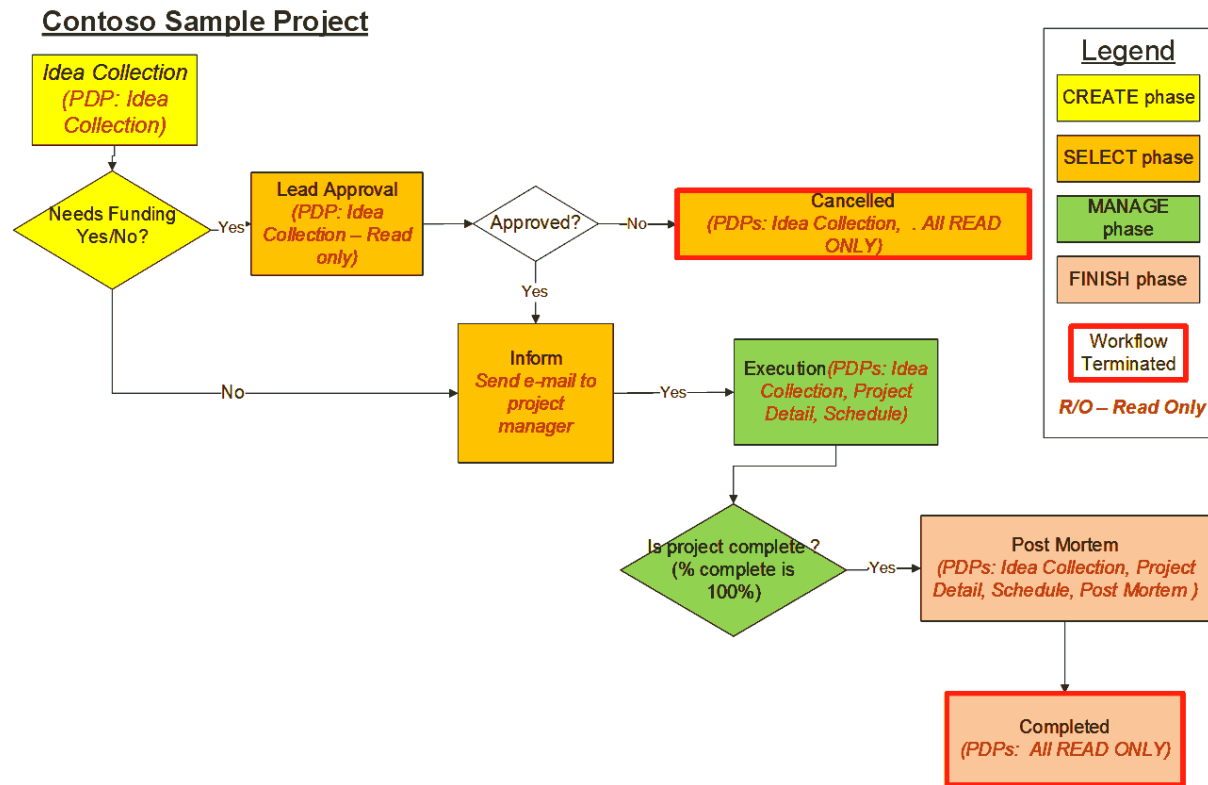


Figure 18: Contoso Sample Process (detailed)

3.5 Objects to configure and use in Project 2013

The following objects are configured and used in Project 2013 for our example:

Objects	List
■ Phases	Create
	Select
	Manage
	Finished

■ Stages (Linked to SharePoint Designer)	Idea Collection
	Lead Approval
	Cancelled
	Inform
	Execution
	Post Mortem
■ Custom Fields and Lookup Tables (Linked to SharePoint Designer)	Proposed Project Manager (Project owner)
	Funding Required
	Proposal Cost
	On Budget
	Reasons
	Lessons Learned
■ Project Detail Pages	Idea Collection (Initial Project Page)
	Post Mortem
■ Default Web Part and Custom Web Parts	Project Fields
■ Enterprise Project Type	Contoso Sample Project
■ Workflow Approval page	Default Project Server Approval

Figure 19: List of objects to configure/use in Project Server 2013

Apart from these workflow-related objects, we need to define some users in the PWA instance.

The workflow uses the members of the Team Lead default security group to get approval on the project when funding is required. At least one user must be a member of this default security group.

4 BUILD/CREATE: WORKFLOW CONFIGURATION

4.1 Enterprise Project Templates

An enterprise project type represents a wrapper that encapsulates phases, stages, a single workflow, and PDPs. Each EPT represents a single project type. Normally, project types are aligned with individual departments, for example, marketing projects, IT projects, or HR projects. Using project types helps to categorize projects within the same organization that have a similar project life cycle. For a user, the EPTs appear in a drop-down list of project types when the user clicks **New Project** on the ribbon in Project Web App.

4.2 Phase: A Collection of Steps in a Project Life Cycle

A phase represents a collection of stages grouped to identify a common set of activities in the project life cycle. Examples of phases are project creation, project selection, and project management (shown as Create, Select, and Manage Phases). Phases do not have any direct technical effect on the behavior of an EPT. That is, changing the order of phases does not affect how the system reacts. The primary purpose of demand management phases is to provide a smoother user experience where users have the option of organizing stages into logical groups.

4.3 Stage: A Step in a Project Life Cycle

A stage represents one step within a project life cycle. A stage is composed of one or more PDPs linked by common logic or theme. Stages at a user level appear as steps within a project. At each step, data must be entered, modified, reviewed, or processed.

4.4 Project Detail Pages (PDPs) in Stages

A PDP represents a single Web Part Page in Project Web App. PDPs can be used to display or collect information from the user. You can create PDPs in much the same way you create any Web Part Page in a SharePoint site, where you can add Web Parts that provide the experience you want. You can add individual Web Parts from the standard Web Part galleries or create custom Web Parts.

Project Server Web Parts and custom Web Parts used in demand management all contain custom fields. Web Parts can make calls to the Project Server Interface (PSI) or the client-side object model (CSOM), query the Reporting database or the **ProjectData** OData service, or integrate with external systems. Figure 20 shows the general hierarchy of the parts of demand management in Project Server 2013.

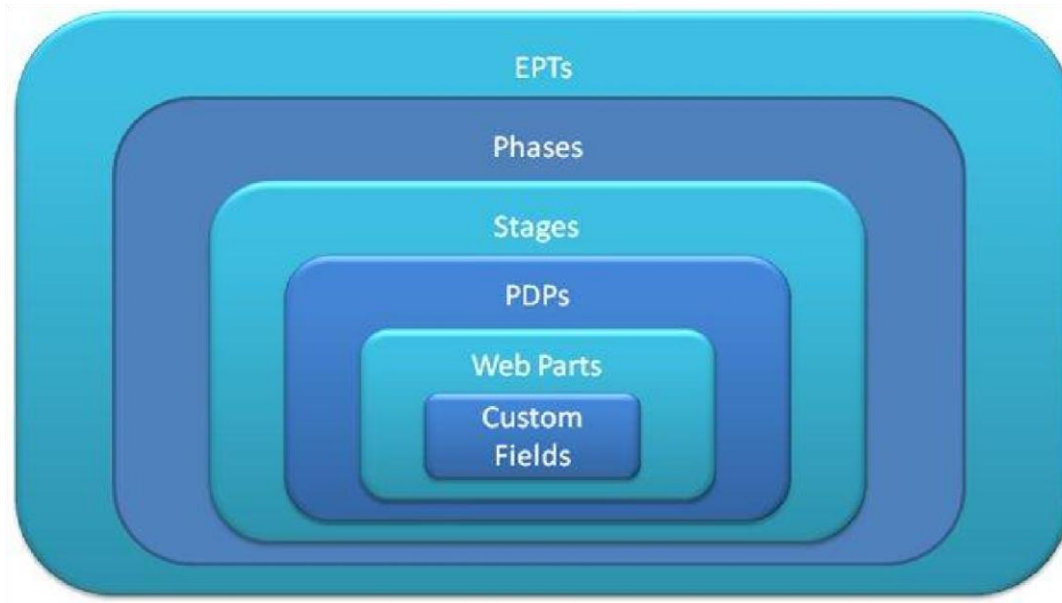


Figure 20: Hierarchy of Project Server objects

4.5 Configuration tasks on Project Server 2013

4.5.1 Initial Setup (On-premises)

The workflow infrastructure for an on-premises installation of Project Server 2013 must be installed with SharePoint Server 2013. It uses a new component named Workflow Manager.

For information about how to install and configure SharePoint 2013 Workflow Manager, see <http://technet.microsoft.com/en-us/library/jj163276>.

4.5.2 Workflow configuration

The following configuration tasks are done on Project Server 2013.

We recommend doing the tasks in this order.

1. Define the list of Lookup Table Values (LTV)
2. Define the list of Custom Fields (CF)
3. Define the list of Project Detail Pages (PDP)
4. Define the list of Workflow Stages
5. Define the list of Workflow Phases
6. Create the enterprise project types (EPT) (without Workflow, if workflow not yet deployed)
7. Define the permissions for users involved with the workflow.

4.6 Configuration Best practices

4.6.1 Naming conventions to use to distinguish specific workflow objects

For each kind of object defined in the workflow configuration, we recommend using specific naming conventions. These conventions ease the initial configuration of the workflow and its maintenance.

Workflow phases:

- Prefix the workflow phase with a number to force the order in the display.
- If a phase is unique for a certain type of Workflow, add an acronym after the number that defines the uniqueness.
- Use specific phases for each workflow, instead of sharing between several different workflows.

For example, for an IT workflow:

1-IT Demand Management	DM
2-IT Portfolio Selection	PS
3-IT Portfolio Planning	PP
4-IT Tracking and Remediation	TR

Figure 21: IT workflow phases

Workflow stages:

- Use a lowercase letter (s, for example) followed by a number, to force the order of display and to distinguish from phases.
- You can also use an acronym after the order letter to attach to a specific workflow.

s1- IT Idea Collection	
s2-IT Lead Approval	
s3-IT Idea Cancelled	

s4-IT Scheduled	
s5- IT Execute	
s6- IT Post Mortem	

Figure 22: IT workflow stages

In our Simple Project example for the phases:

1-Create	CR
2-Select	SE
3-Manage	MA
4-Finished	FI

Figure 23: Contoso Sample Project example phases

In our Simple Project example for the stages:

s1-Idea Collection	ic
s2-Lead Approval	ap
s3-Inform	in
s4-Cancelled	ca
s5-Execution	ex
s6-Post Mortem	pm

Figure 24: Contoso Sample Project example stages

4.6.2 List of custom fields used in our example

In our Simple Project example we use the following fields:

Project Name (default field)
Description (default field)
Funding Required
Proposal Cost
Proposed Project Manager (=> Project Owner, default field)
Lessons Learned
On Budget
Reasons

Figure 25: Contoso list of custom fields

4.7 How to create workflow objects in Project Online

This chapter describes the steps involved to create the different objects of our example using the administrative pages of Project Web App.

We are using an online tenant. <https://msftepm.sharepoint.com/sites/pwa>

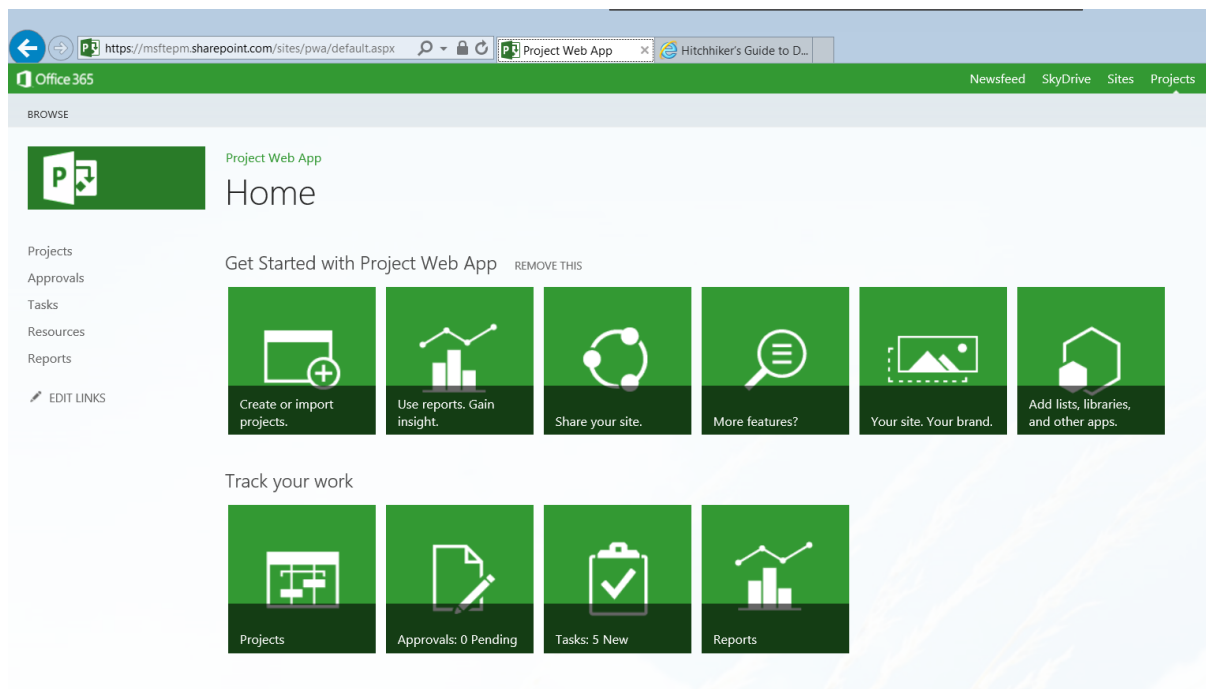


Figure 26: Project Web App Online tenant

We show how to create the following objects:

1. Custom fields and lookup tables
2. Web Parts
3. Project Detail Pages (PDPs)
4. Workflow phases
5. Workflow stages
6. Enterprise project type (EPT)

A Project Web App instance no longer includes a sample workflow installed by default, as in Project Server 2010.

You must log on using the Functional Project administrator, to do the configuration.

Create the Project Server objects in a bottom-up order, following the hierarchy presented in Figure 20: Hierarchy of Project Server objects.

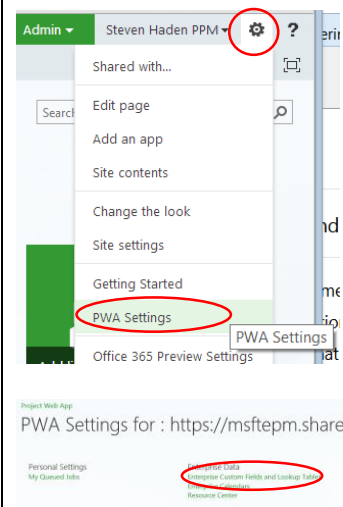
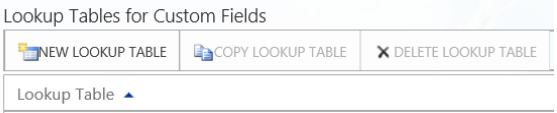
4.7.1 Lookup tables and custom fields

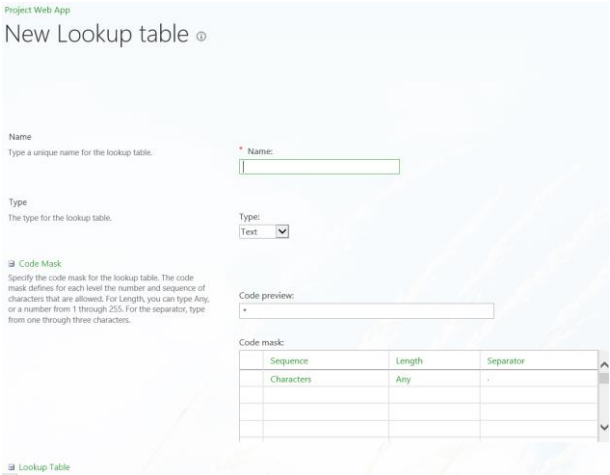
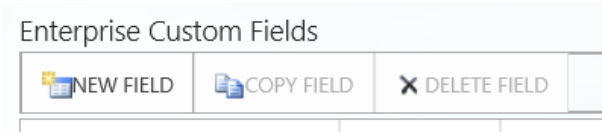
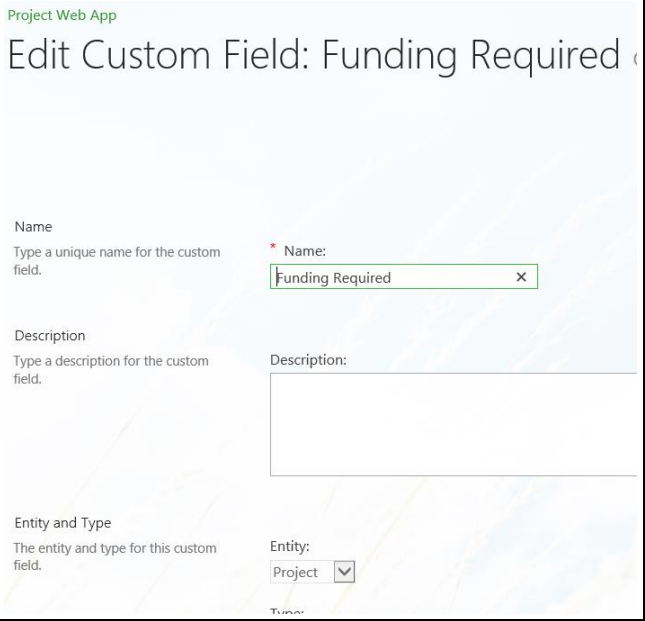
In this chapter we first create the lookup tables (LT) and then the custom fields for our example.

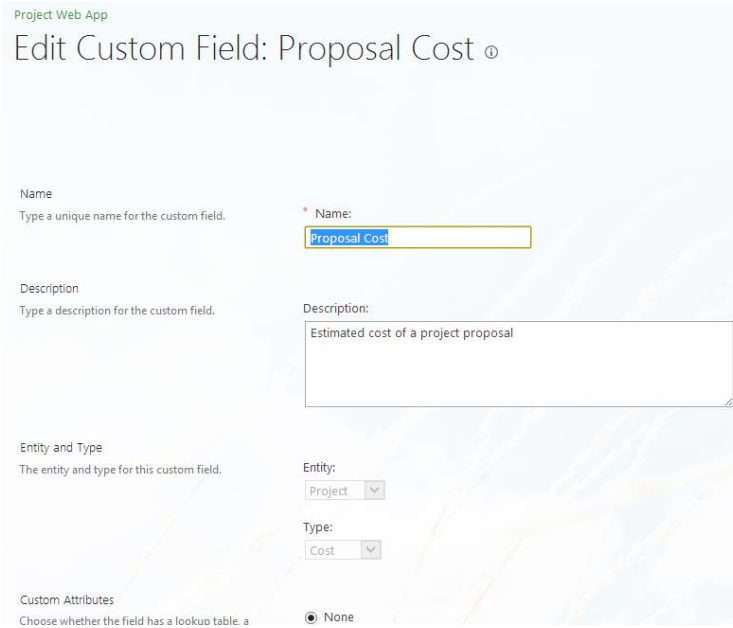
Field name	Custom Field (CF) or Lookup Table (LT)	Type
Funding Required LT	LT	Text: Value Yes/No
Funding Required	CF	Project of Type Text Using a lookup table
Proposal Cost	CF	Project of Type Currency
Lessons Learned	CF	Project of Type Text
On Budget LT	LT	Text: Value OK/Under/Exceeded

On Budget	CF	Project of Type Text Using a lookup table
Reasons	CF	Project of Type Text

Figure 27: List of CF and LT for Contoso sample

Actions	Screen
In Project Web App, after selecting the Settings icon in the upper right corner select PWA Server Settings , and then select Enterprise Custom Fields and Lookup Tables	 A screenshot of the Project Web App (PWA) settings interface. At the top, there is a navigation bar with 'Admin' and 'Steven Haden PPM'. A settings gear icon is circled in red. Below it, a dropdown menu is open, showing options like 'Shared with...', 'Edit page', 'Add an app', 'Site contents', 'Change the look', 'Site settings', 'Getting Started', 'PWA Settings' (circled in red), and 'Office 365 Preview Settings'. Below the menu, the URL 'PWA Settings for : https://msftepm.sharepoint.com/sites/pwa' is shown. At the bottom, there are several links: 'Personal Settings: My Queue Jobs', 'Enterprise Custom Fields and Lookup Tables' (circled in red), and 'Queue and Database Administration: Manage Queue Jobs, Delete Enterprise Objects, Force Check-in Enterprise Objects'.
Select New Lookup Tables	 A screenshot of the 'Lookup Tables for Custom Fields' page. It features three buttons: 'NEW LOOKUP TABLE', 'COPY LOOKUP TABLE', and 'DELETE LOOKUP TABLE'. Below these buttons is a search bar labeled 'Lookup Table' with a dropdown arrow.

Actions	Screen
<p>Name: Funding required</p> <p>Type: Text</p> <p>Code Mask: *</p> <p>Lookup table values:</p> <p>Yes</p> <p>No</p>	
<p>To create the “Funding Required” field, select New Field in the Enterprise Custom Fields section.</p> <p>Select the Field button</p>	
<p>Name: Funding required</p> <p>Entity and Type: Project</p> <p>Text</p> <p>Custom Attributes:</p> <p>Lookup Table: Funding Required</p> <p>Behavior: Select Behavior controlled by workflow</p>	

Actions	Screen
<p>Name: Proposal Cost</p> <p>Entity and Type: Project</p> <p>Type: Cost</p> <p>Behavior: Select Behavior controlled by workflow</p>	 <p>Project Web App</p> <h3>Edit Custom Field: Proposal Cost ⓘ</h3> <p>Name Type a unique name for the custom field.</p> <p>* Name: <input type="text" value="Proposal Cost"/></p> <p>Description Type a description for the custom field.</p> <p>Description: <input type="text" value="Estimated cost of a project proposal"/></p> <p>Entity and Type The entity and type for this custom field.</p> <p>Entity: <input type="text" value="Project"/></p> <p>Type: <input type="text" value="Cost"/></p> <p>Custom Attributes Choose whether the field has a lookuo table. a</p> <p><input checked="" type="radio"/> None</p>

Actions	Screen
<p>Name: Lessons Learned</p> <p>Entity and Type: Project Text</p> <p>Custom Attributes: Multiple Lines of Text</p> <p>Behavior: Select Behavior controlled by workflow</p>	<div><h3>Edit Custom Field: Lessons Learned ⓘ</h3><div><div><div>Name</div><div>Type a unique name for the custom field.</div></div><div><div>* Name:</div><div>Lessons Learned</div></div></div><div><div>Description</div><div>Type a description for the custom field.</div></div><div><div>Description:</div><div></div></div><div><div>Entity and Type</div><div>The entity and type for this custom field.</div></div><div><div>Entity:</div><div>Project</div></div><div><div>Type:</div><div>Text</div></div><div><div>Custom Attributes</div><div>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.</div></div><div><div><div><input type="radio"/> Single line of text</div><div><input checked="" type="radio"/> Multiple lines of text</div><div><input type="radio"/> Lookup Table</div><div><input type="radio"/> Formula</div></div></div></div>

Actions	Screen																											
<p>Lookup table</p> <p>Name: On Budget</p> <p>Type: Text</p> <p>Code Mask: *</p> <p>Lookup table values:</p> <p>OK</p> <p>Under</p> <p>Exceeded</p>	<div><div>Project Web App</div><div>Search this</div><h2>Edit Lookup table: On Budget</h2><div><div><div>Name</div><div>Type a unique name for the lookup table.</div><div><div>On Budget</div></div></div><div><div>Type</div><div>The type for the lookup table.</div><div><div>Type:</div><div>Text</div></div></div><div><div>Code Mask</div><div>Specify the code mask for the lookup table. The code mask defines for each level the number and sequence of characters that are allowed. For Length, you can type Any, or a number from 1 through 255. For the separator, type from one through three characters.</div></div><div><div>Code preview:</div><div>*</div></div><div><div>Code mask:</div><table><thead><tr><th>Sequence</th><th>Length</th><th>Separator</th></tr></thead><tbody><tr><td>Characters</td><td>Any</td><td>.</td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></tbody></table></div><div><div>Lookup Table</div><div>Edit the lookup table</div><div><div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div><div><table><thead><tr><th>Level</th><th>Value</th><th>Description</th></tr></thead><tbody><tr><td>1</td><td>OK</td><td></td></tr><tr><td>1</td><td>Under</td><td></td></tr><tr><td>1</td><td>Exceeded</td><td></td></tr></tbody></table></div></div></div></div></div></div>	Sequence	Length	Separator	Characters	Any	.										Level	Value	Description	1	OK		1	Under		1	Exceeded	
Sequence	Length	Separator																										
Characters	Any	.																										
Level	Value	Description																										
1	OK																											
1	Under																											
1	Exceeded																											

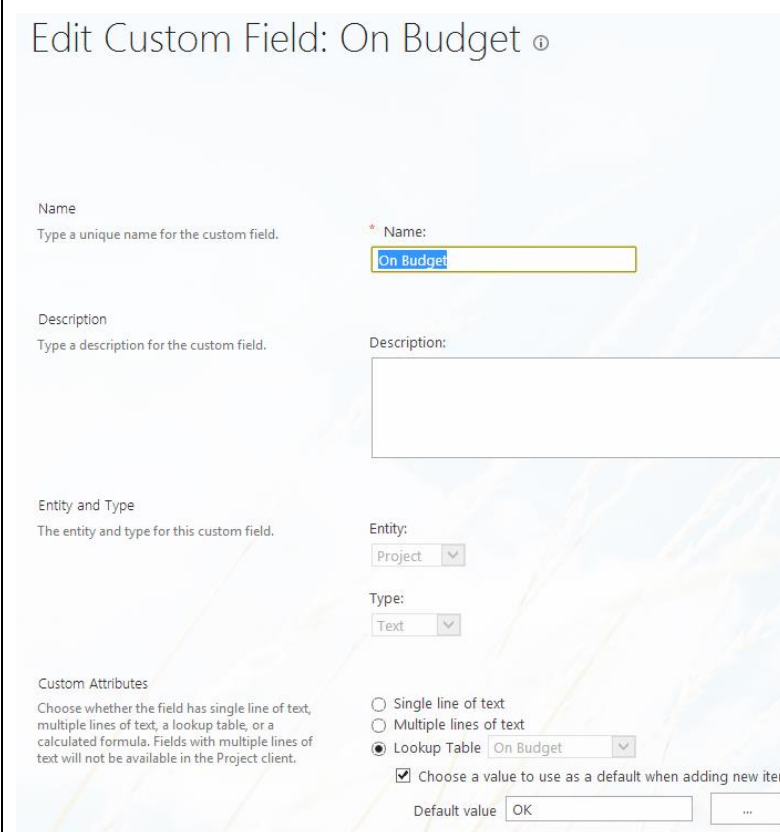
Actions	Screen
<p>Name: On Budget</p> <p>Entity and Type:</p> <p>Project</p> <p>Text</p> <p>Custom Attributes: Single Line of Text</p> <p>Behavior: Select Behavior controlled by workflow</p>	 <p>Figure 28 shows the 'Edit Custom Field: On Budget' configuration screen. The screen is divided into several sections: 'Name' (with a unique name field set to 'On Budget'), 'Description' (with a text area), 'Entity and Type' (with 'Project' selected for Entity and 'Text' selected for Type), and 'Custom Attributes' (with 'Single line of text' selected). There is also a checkbox for 'Choose a value to use as a default when adding new item' and a 'Default value' field set to 'OK'.</p>

Figure 28: Steps to create CF and LT

4.7.2 Project Detail Pages Creation

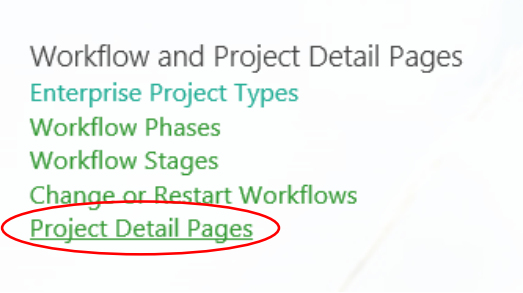
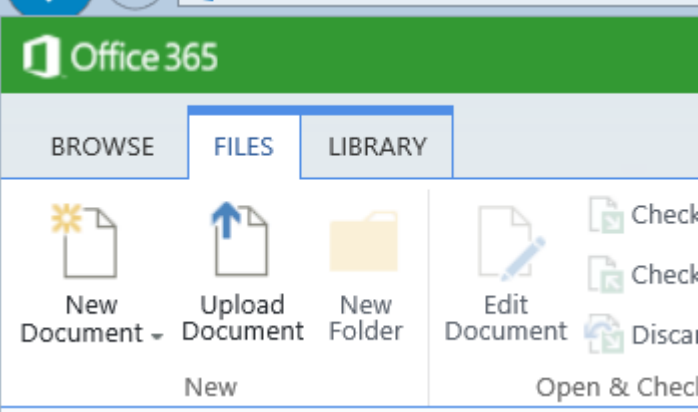
There are three types of PDPs that can be created:

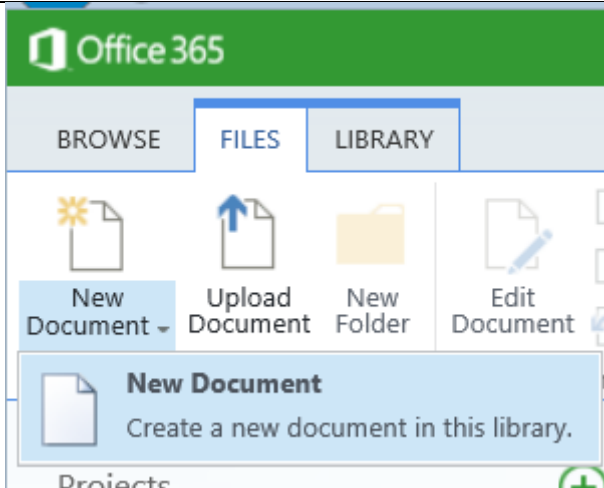
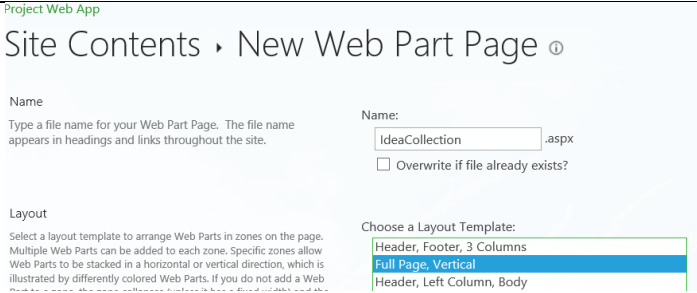
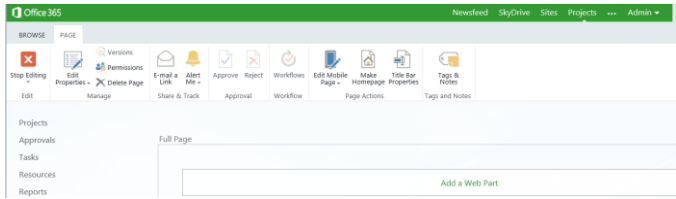
- **New Project:** Used for creating a project. This type of PDP is required with an enterprise project template that has a workflow for portfolio analysis.
- **Workflow Status:** Shows the current stage and status for a project proposal.
- **Project:** Used for editing project details in a non-workflow enterprise project template, or in other applications.

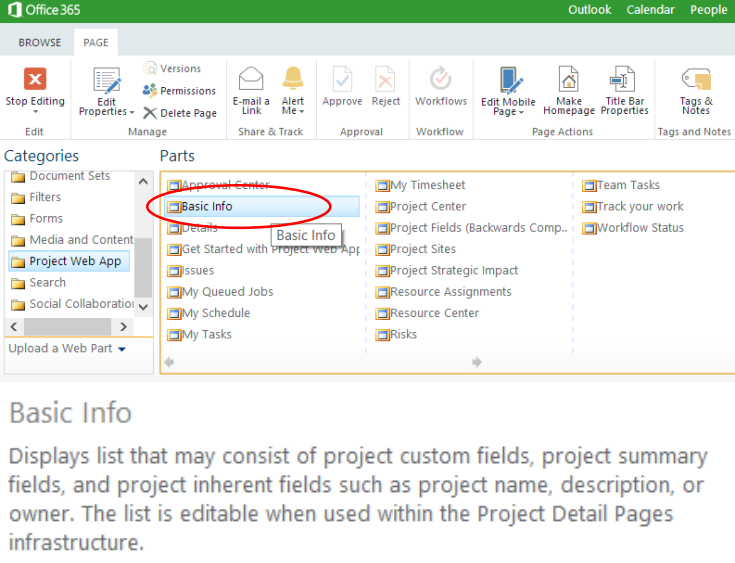
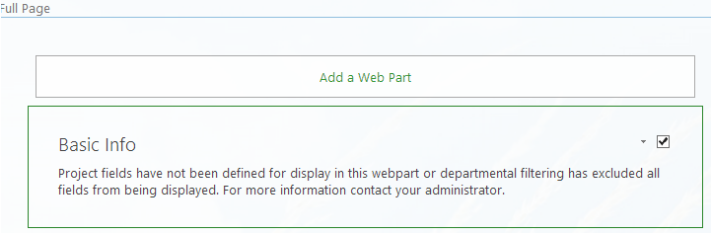
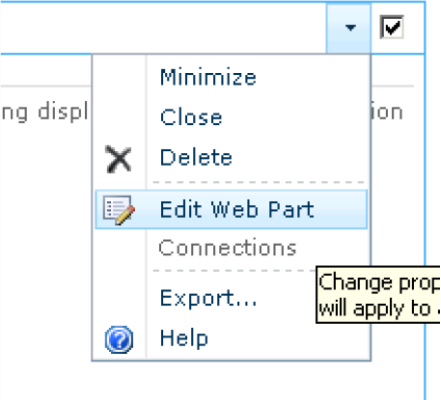
It is a good practice to create your own New Project page, so that you can directly enter required information when creating your project.

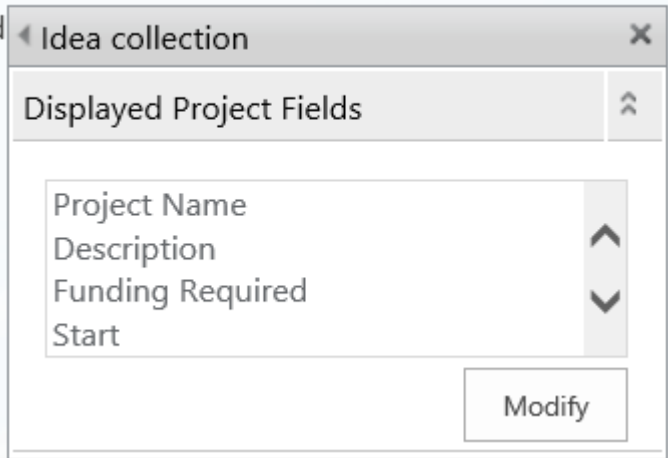
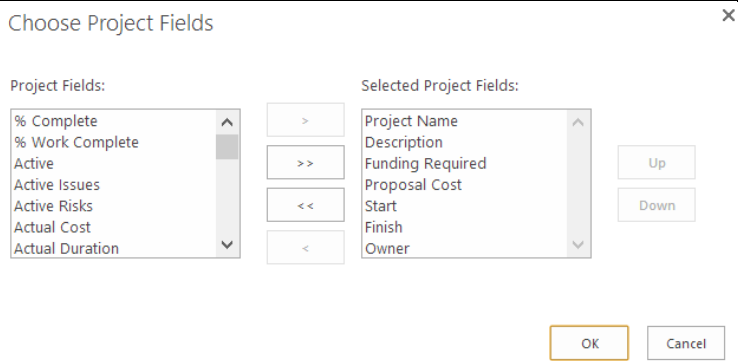
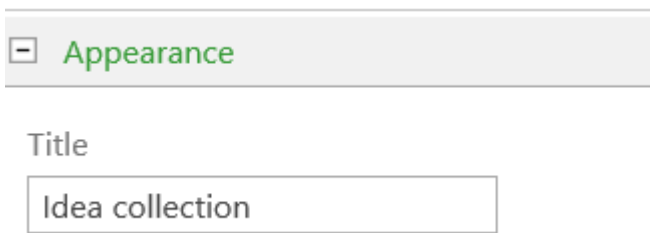
In our example we create the following PDPs:

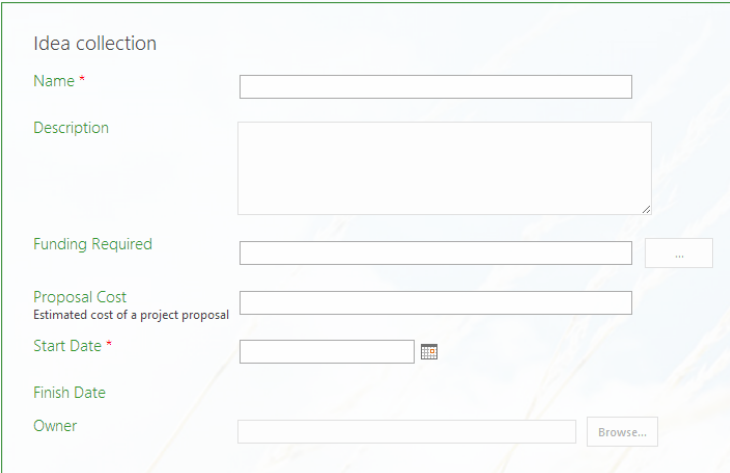
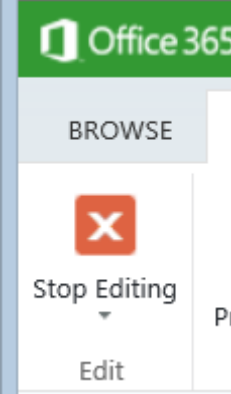
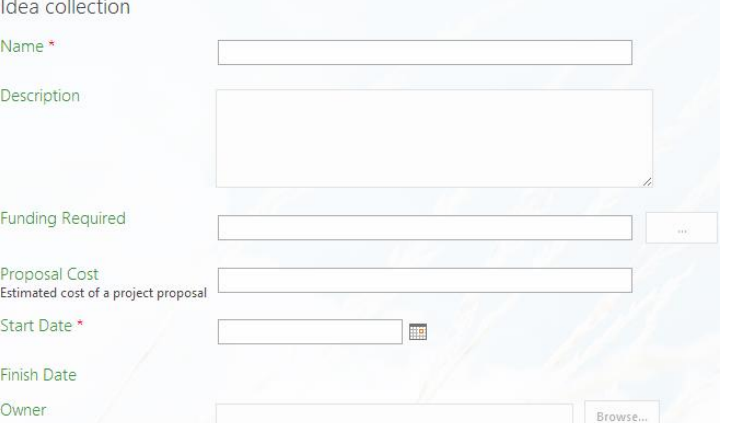
- IdeaCollection
- PostMortem

Actions	Screen
<p>In Project Web App, after selecting Settings/Server Settings:</p> <p>Select Project Detail Pages in the last section</p>	 <p>Workflow and Project Detail Pages</p> <ul style="list-style-type: none"> Enterprise Project Types Workflow Phases Workflow Stages Change or Restart Workflows Project Detail Pages
<p>Select the FILES tab in the ribbon</p>	 <p>Office 365</p> <p>BROWSE FILES LIBRARY</p> <p>New Document Upload Document New Folder Edit Document Check Check Discal Open & Chec</p>

Actions	Screen
<p>Select New Document in the ribbon</p>	
<p>Select Full Page, Vertical and name the page:</p> <p>IdeaCollection</p>	
<p>We can now add some Web Parts to this blank page.</p> <p>Select Add a Web Part</p>	

Actions	Screen
<p>In the Categories section, select: Project Web App</p> <p>In the Parts section, select: Basic Info</p> <p>Click Add</p>	 <p>Basic Info</p> <p>Displays list that may consist of project custom fields, project summary fields, and project inherent fields such as project name, description, or owner. The list is editable when used within the Project Detail Pages infrastructure.</p>
<p>To populate the Web Part, add the project fields.</p>	 <p>Full Page</p> <p>Add a Web Part</p> <p>Basic Info</p> <p>Project fields have not been defined for display in this webpart or departmental filtering has excluded all fields from being displayed. For more information contact your administrator.</p>
<p>Select Edit Web Part in the right top menu.</p>	 <p>Minimize</p> <p>Close</p> <p>Delete</p> <p>Edit Web Part</p> <p>Connections</p> <p>Export...</p> <p>Help</p> <p>Change properties will apply to .</p>

Actions	Screen
Select Modify	
Select the following fields: Project Name Description Funding Required Proposal Cost Start Finish Owner	
Change the title of the page to: Idea Collection Select OK to close the Web Part editor.	

Actions	Screen
The PDP is now ready to save.	
Select the Page Tools tab in the ribbon, and then choose Stop Editing .	
The PDP page is now finished.	

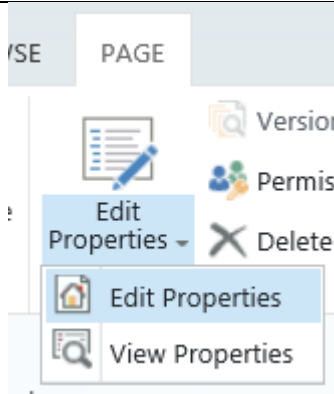
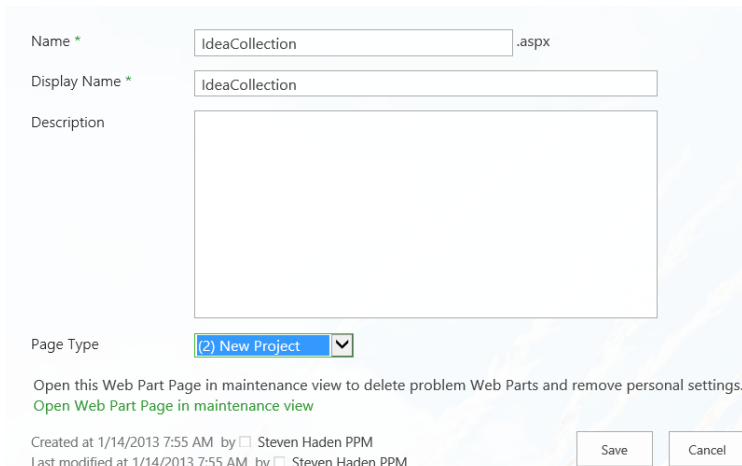
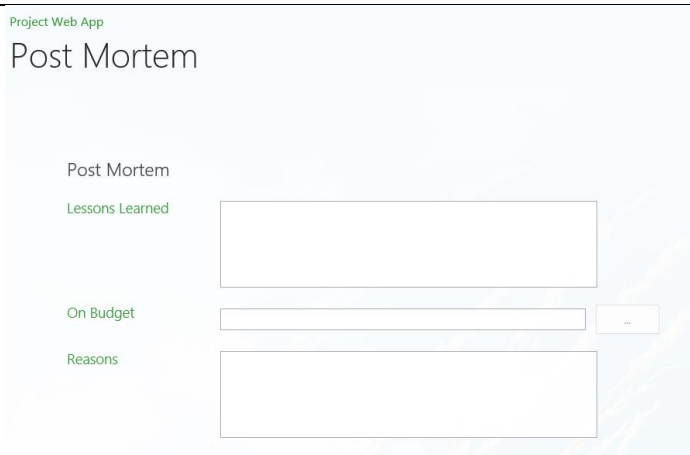
Actions	Screen
On the PAGE tab, choose Edit Properties .	
Change the Display Name to: Idea Collection For the Page Type , select New Project .	
Do the same for the Post Mortem page. Add these fields:	

Figure 29: Steps to create PDPs

4.7.3 Workflow Phases Creation

The following phases will be created:

- 1-Create

- 2-Select
- 3-Manage
- 4-Finished

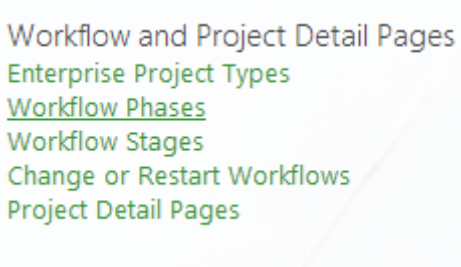
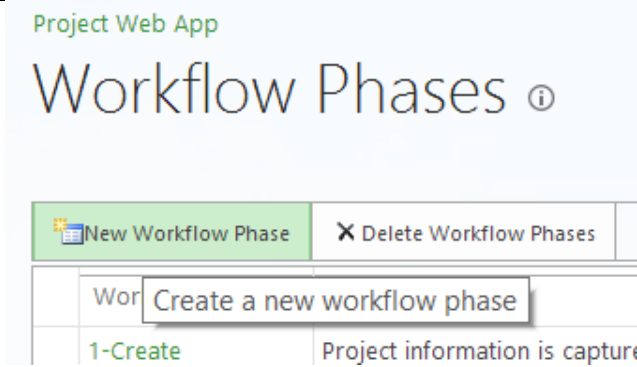
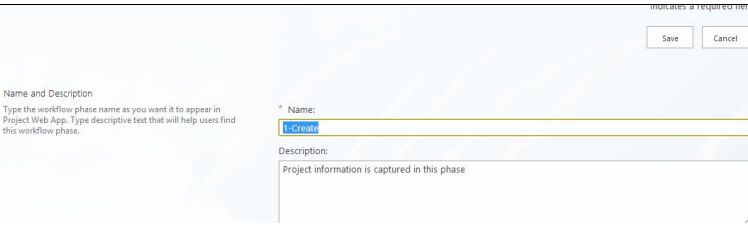
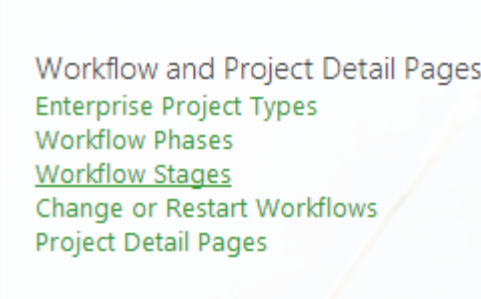
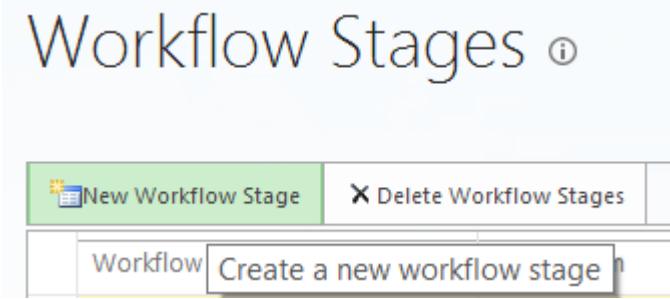
Actions	Screen										
<p>In Project Web App, after selecting Settings/Server Settings:</p> <p>Select Workflow Phases</p>											
Select New Workflow Phase											
<p>Name :</p> <p>1-Create</p> <p>Add a description.</p>											
<p>Create the following phases:</p> <p>2-Select</p> <p>3-Manage</p> <p>4-Finished</p>	<table border="1"> <thead> <tr> <th>Workflow Phases</th><th>Description</th></tr> </thead> <tbody> <tr> <td>1-Create</td><td>Project information is captured in this phase</td></tr> <tr> <td>2-Select</td><td>A subset of projects are selected in this phase</td></tr> <tr> <td>3-Manage</td><td>Projects in execution are monitored in this phase</td></tr> <tr> <td>4-Finished</td><td>The workflow is finished</td></tr> </tbody> </table>	Workflow Phases	Description	1-Create	Project information is captured in this phase	2-Select	A subset of projects are selected in this phase	3-Manage	Projects in execution are monitored in this phase	4-Finished	The workflow is finished
Workflow Phases	Description										
1-Create	Project information is captured in this phase										
2-Select	A subset of projects are selected in this phase										
3-Manage	Projects in execution are monitored in this phase										
4-Finished	The workflow is finished										

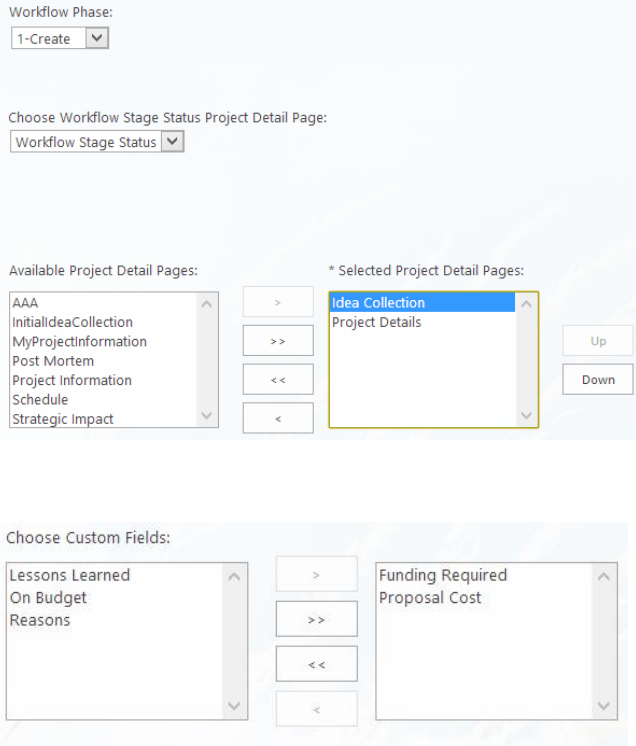
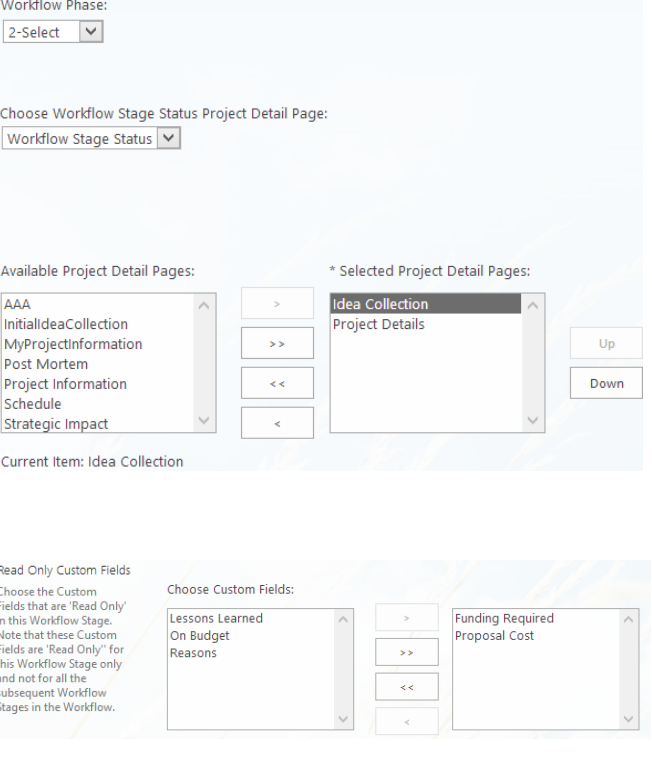
Figure 30: Steps to create phases

4.7.4 Workflow Stages Creation

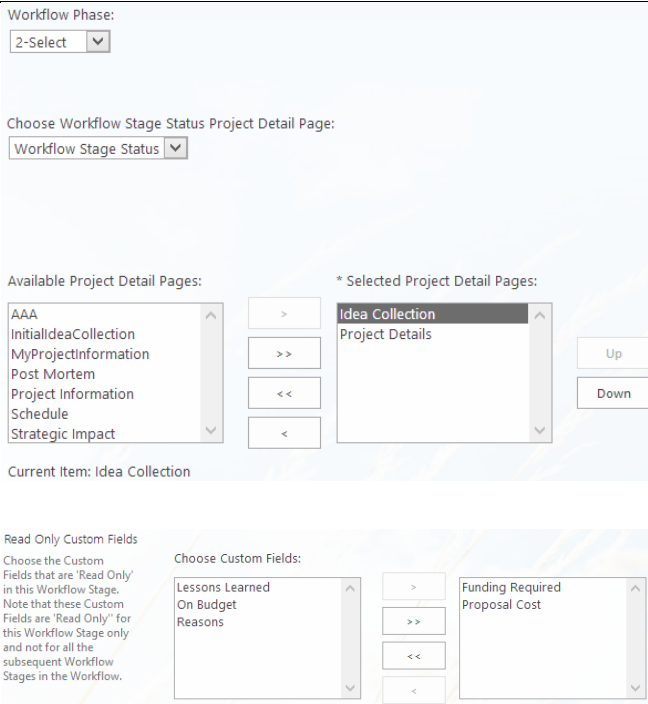
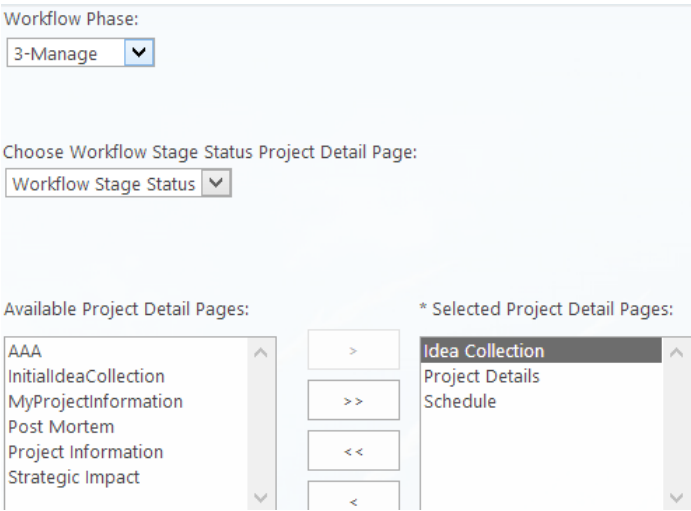
The following five stages are created:


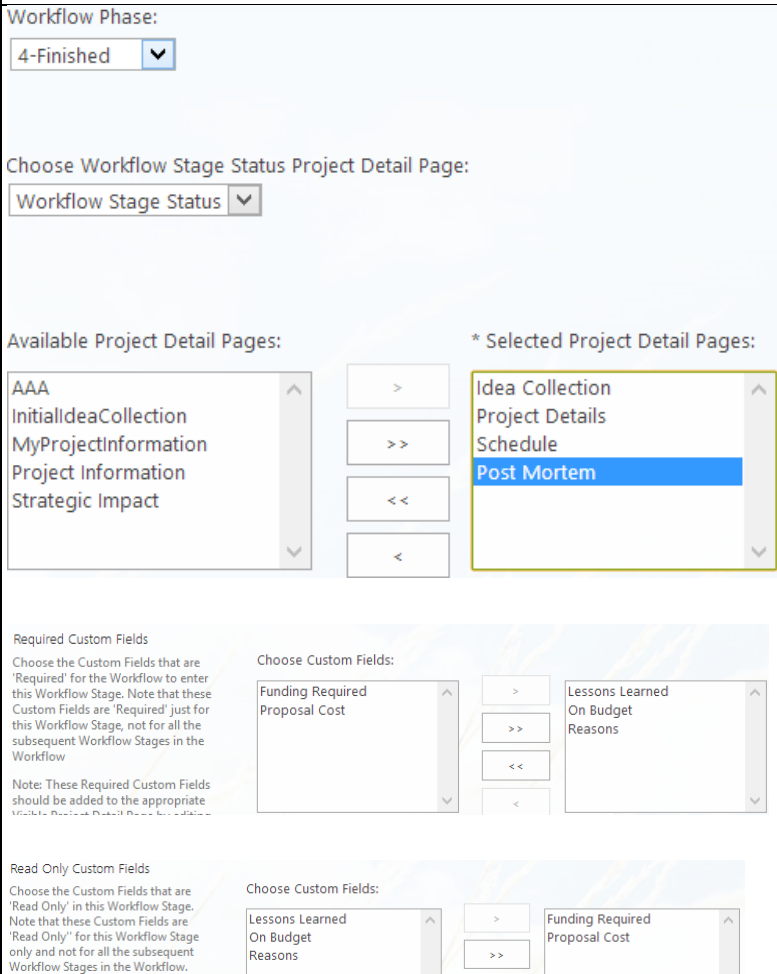
- s1-Idea Collection
- s2-Lead Approval
- s3-Inform
- s4-Cancelled
- s5-Execution
- s6-Post Mortem

Actions	Screen
<p>In Project Web App, after selecting Settings/Server Settings:</p> <p>Select Workflow Stages</p>	
<p>Select New Workflow Stage</p>	

Actions	Screen
<p>Enter the following</p> <p>Name: s1- Idea Collection</p> <p>Description</p> <p>Workflow Phase: 1-Create</p> <p>Selected Project Detail Pages:</p> <p>Idea Collection Project Details</p> <p>Required custom fields:</p> <p>Funding Required Proposal Cost</p>	
<p>Name: s2- Lead Approval</p> <p>Description</p> <p>Workflow phase: 2-Select</p> <p>Selected Project Detail Pages:</p> <p>Idea Collection Project Details</p> <p>Required custom fields:</p> <p>Funding required (read-only) Proposal Cost(read-only)</p>	

Actions	Screen
<p>Name: s3- Inform</p> <p>Description</p> <p>Workflow phase: 2-Select</p> <p>Selected Project Detail Pages: Idea Collection Project Details</p> <p>Required custom fields: Funding Required (read-only) Proposal Cost</p>	<div><p>Workflow Phase: 2-Select</p><p>Choose Workflow Stage Status Project Detail Page: Workflow Stage Status</p><div><p>Available Project Detail Pages:</p><div>AAA InitialIdeaCollection MyProjectInformation Post Mortem Project Information Schedule Strategic Impact</div><div>> >> << <</div><p>* Selected Project Detail Pages:</p><div>Idea Collection Project Details</div><div>Up Down</div><p>Current Item: Idea Collection</p></div><div><p>Read Only Custom Fields</p><p>Choose the Custom Fields that are 'Read Only' in this Workflow Stage. Note that these Custom Fields are 'Read Only' for this Workflow Stage only and not for all the subsequent Workflow Stages in the Workflow.</p><div><p>Choose Custom Fields:</p><div>Lessons Learned On Budget Reasons</div><div>> >> << <</div><div>Funding Required Proposal Cost</div></div></div></div>

Actions	Screen
Name: s4- Cancelled Description Workflow phase: 2-Select Selected Project Detail Pages: Idea Collection Project Details Required custom fields: Funding required (read-only) Proposal Cost(read-only)	
Name: s5- Execution Description Workflow phase: 3-Manage Selected Project Detail Pages: Idea Collection Project Details	

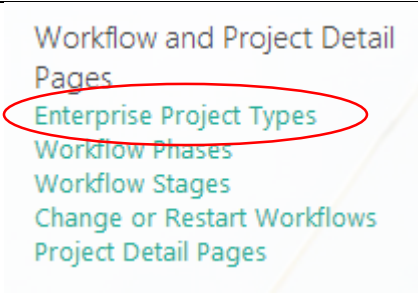
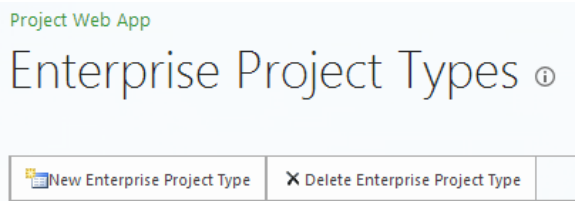
Actions	Screen
Required custom fields: Funding required (read-only) Proposal Cost(read-only)	
Name: s5- Post Mortem Description Workflow phase: 4-Finished Selected Project Detail Pages: Post Mortem Idea Collection Project Details Schedule Required custom fields: Lessons Learned On Budget Reasons Read-only fields: Proposal Cost Funding required	

Actions	Screen																																												
Result of the stages creation	<table><tr><th>Workflow Stage</th><th>Description</th><th>Visible Project Detail Pages</th><th>Required Custom Fields</th></tr><tr><td>Workflow Phase Name: 1- Create</td><td></td><td></td><td></td></tr><tr><td>s1-IdeaCollection</td><td></td><td>Idea Collection, Project Detail</td><td>Funding Required, Proposal</td></tr><tr><td>Workflow Phase Name: 2- Select</td><td></td><td></td><td></td></tr><tr><td>s2-LeadApproval</td><td></td><td>Project Details</td><td></td></tr><tr><td>s3-Inform</td><td>inform the Project Manager</td><td>Idea Collection</td><td></td></tr><tr><td>s4-Cancelled</td><td></td><td>Idea Collection, Project Detail</td><td></td></tr><tr><td>Workflow Phase Name: 3- Manage</td><td></td><td></td><td></td></tr><tr><td>s5-Execution</td><td></td><td>Project Details, Schedule</td><td></td></tr><tr><td>Workflow Phase Name: 4- Finished</td><td></td><td></td><td></td></tr><tr><td>s6-PostMortem</td><td></td><td>Idea Collection, Schedule, PoI</td><td>Lessons Learned, On Budget</td></tr></table>	Workflow Stage	Description	Visible Project Detail Pages	Required Custom Fields	Workflow Phase Name: 1- Create				s1-IdeaCollection		Idea Collection, Project Detail	Funding Required, Proposal	Workflow Phase Name: 2- Select				s2-LeadApproval		Project Details		s3-Inform	inform the Project Manager	Idea Collection		s4-Cancelled		Idea Collection, Project Detail		Workflow Phase Name: 3- Manage				s5-Execution		Project Details, Schedule		Workflow Phase Name: 4- Finished				s6-PostMortem		Idea Collection, Schedule, PoI	Lessons Learned, On Budget
Workflow Stage	Description	Visible Project Detail Pages	Required Custom Fields																																										
Workflow Phase Name: 1- Create																																													
s1-IdeaCollection		Idea Collection, Project Detail	Funding Required, Proposal																																										
Workflow Phase Name: 2- Select																																													
s2-LeadApproval		Project Details																																											
s3-Inform	inform the Project Manager	Idea Collection																																											
s4-Cancelled		Idea Collection, Project Detail																																											
Workflow Phase Name: 3- Manage																																													
s5-Execution		Project Details, Schedule																																											
Workflow Phase Name: 4- Finished																																													
s6-PostMortem		Idea Collection, Schedule, PoI	Lessons Learned, On Budget																																										

Figure 31: Steps to create stages

4.7.5 Enterprise Project Type creation

Create the enterprise project type (EPT): Contoso Project.

Actions	Screen
<p>In Project Web App, after selecting PWA Settings:</p> <p>In the Workflow and Project Detail Pages section</p> <p>select: Enterprise Project Types</p>	
<p>Select New Enterprise Project Type</p>	

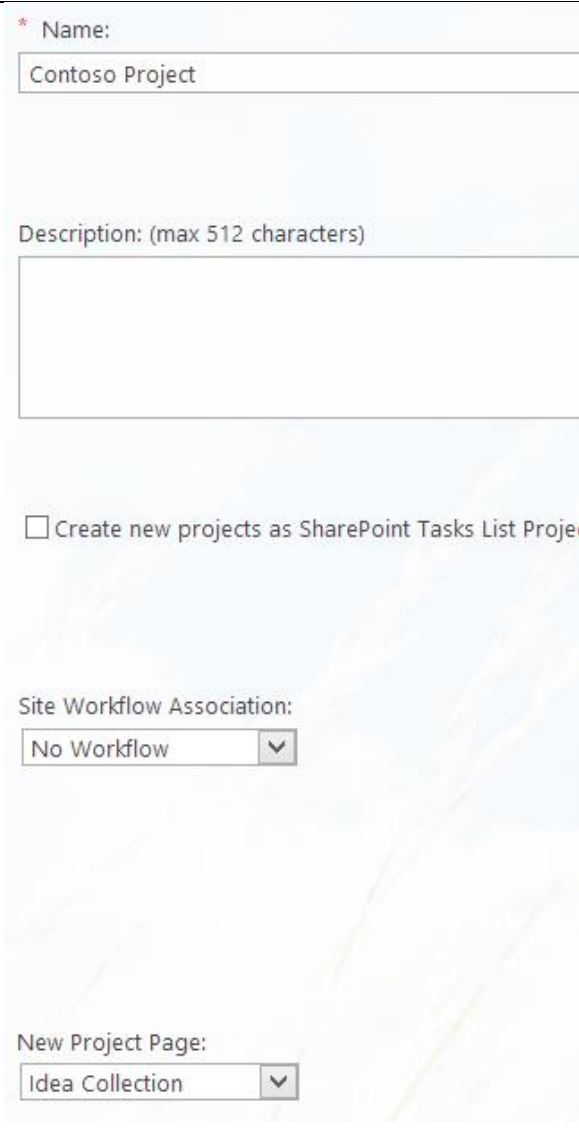
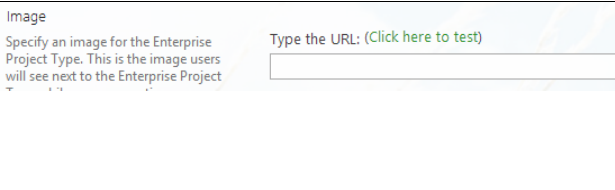
Actions	Screen
<p>Name: Contoso Project</p> <p>Description:</p> <p>Site Workflow Association: No workflow</p> <p>New Project Page: Idea Collection</p> <p>It is not necessary to select Project Detail Pages.</p>	
<p>If you want to specify an image with this EPT, you can specify a URL in a shared document library containing the image</p>	

Figure 32: Steps to create an EPT

To be able to choose the workflow associated with this EPT, the workflow must be deployed in the Site Collection of Project Web App (see chapter 5.9).

5 WORKFLOW ORCHESTRATION: THE BIG PICTURE

The steps described in this chapter can be found in the following blog article:

<http://blogs.office.com/b/project/archive/2012/09/18/demand-management-sharepoint-designer-project-server.aspx>

Here are also some good sites with more information:

- Getting started developing Project Server 2013 workflows:
<http://msdn.microsoft.com/en-us/library/ee767694?v=office15.aspx>
- Workflow development with SharePoint Designer or Visio 2013
[http://msdn.microsoft.com/en-us/library/jj163272?v=\(office.15\).aspx](http://msdn.microsoft.com/en-us/library/jj163272?v=(office.15).aspx)
- Develop SharePoint 2013 workflows using Visual Studio
[http://msdn.microsoft.com/en-us/library/jj163199\(v=office.15\).aspx](http://msdn.microsoft.com/en-us/library/jj163199(v=office.15).aspx)

5.1 Creating Project 2013 workflows

Figure 33 shows a high-level view of the steps to create project workflows, from the first steps in Project Web App to the final steps:

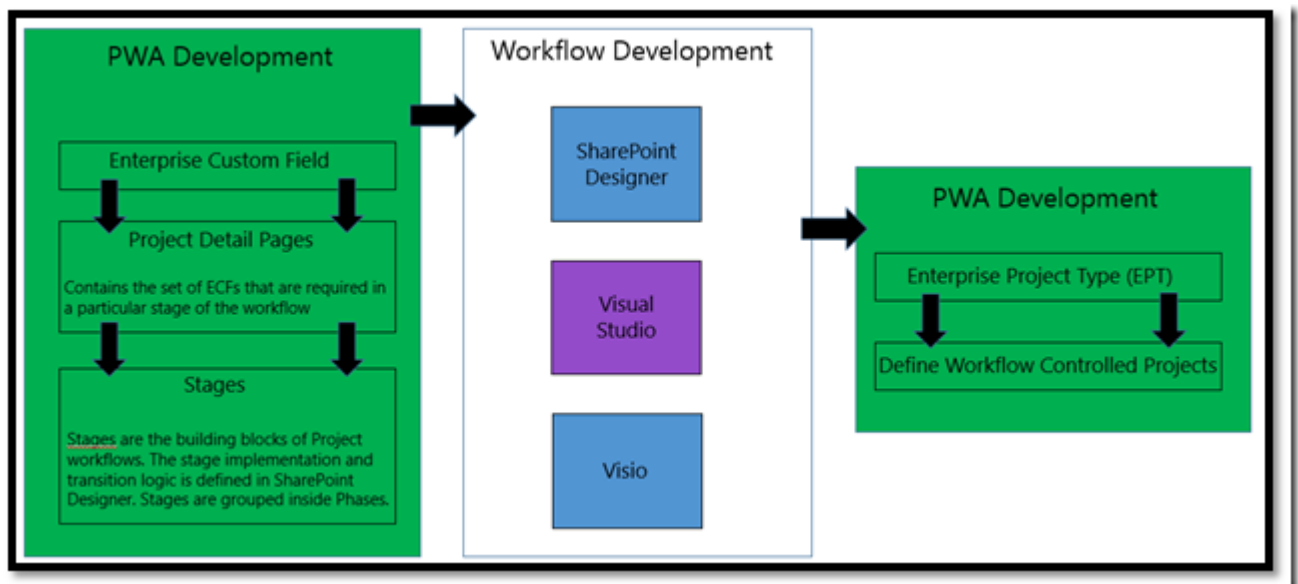


Figure 33: High level steps to create a Project Server 2013 workflow

Visio 2013 and SharePoint Designer 2013 enable business analysts, process consultants, and IT professionals to collaborate and build workflows. Both Visio Professional 2013 and the Visual Designer in SharePoint Designer 2013 provide a rich representation of workflows in a format that is understandable to programmers and non-programmers alike.

5.2 Workflow Key concepts

Project 2013 workflows are built from SharePoint 2013 Site Workflows.

SharePoint Workflows are based on Windows Workflow Foundation, which in turn is built with the .NET Framework v4.

Windows Workflow Foundation 4.0 (WF4) is substantially redesigned from earlier versions. WF4, in turn, is built on the messaging functionality provided by WCF. SharePoint 2013 treats workflow as a separate service and leverages Windows Azure Workflow, which includes WF4. The workflow service provided by Windows Azure Workflow is decoupled from SharePoint and no longer runs in the SharePoint farm, rather it runs on its own servers. Windows Azure Workflow also leverages the latest and greatest advancements in workflow capabilities, performance and scalability from Microsoft.

SharePoint 2013 Workflow Architecture

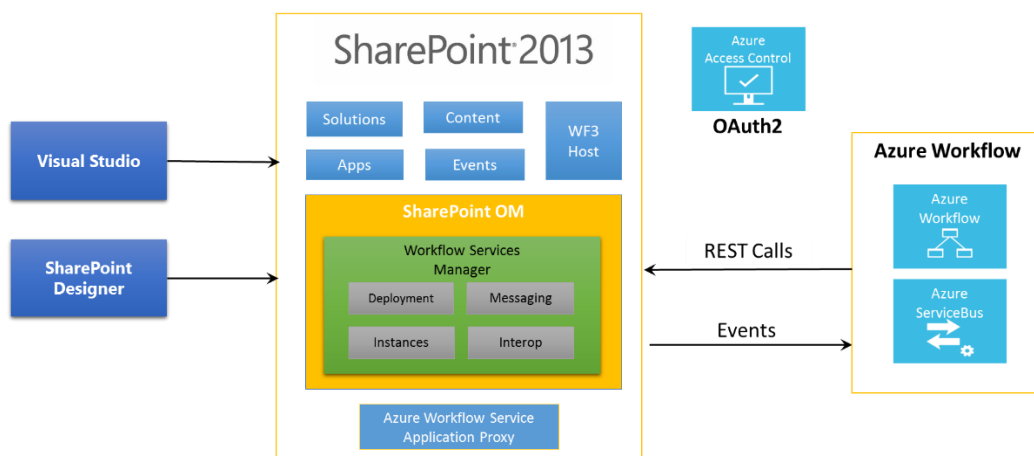


Figure 34: SharePoint 2013 Workflow Architecture

Another of the prominent changes is that workflows on the WF 4 platform are fully declarative. That is, workflows are no longer compiled into managed assemblies and deployed to an assembly cache. Instead, XAML files define your workflows and frame their execution.

SharePoint Designer 2013 has been updated with the goal of making it the authoring environment of choice for authoring SharePoint workflows. SharePoint Designer 2013 provides workflow authors with both a designer surface and a text-based workflow authoring environment. Additionally, you

can develop workflow custom actions in Visual Studio 2012 and then import them into SharePoint Designer 2013, where they can then be accessed from the Workflow Designer.

5.3 Workflow Creation, Administration, and Usage Process

Figure 35 shows the processes involved in the creation, administration, and usage of project workflow. It also shows the actors involved.

1. The Project Server workflow objects are created by the PWA administrator.
2. The workflow is created in SharePoint Designer 2013 by a business analyst or a developer.
3. The workflow is deployed in Project Web App instance by a SharePoint/Project administrator.
4. The PWA administrator associates the workflow with a project EPT.
5. A user creates a new project and initiates the workflow.
6. The workflow starts and waits for user input at the different defined stages.
7. The workflow terminates.

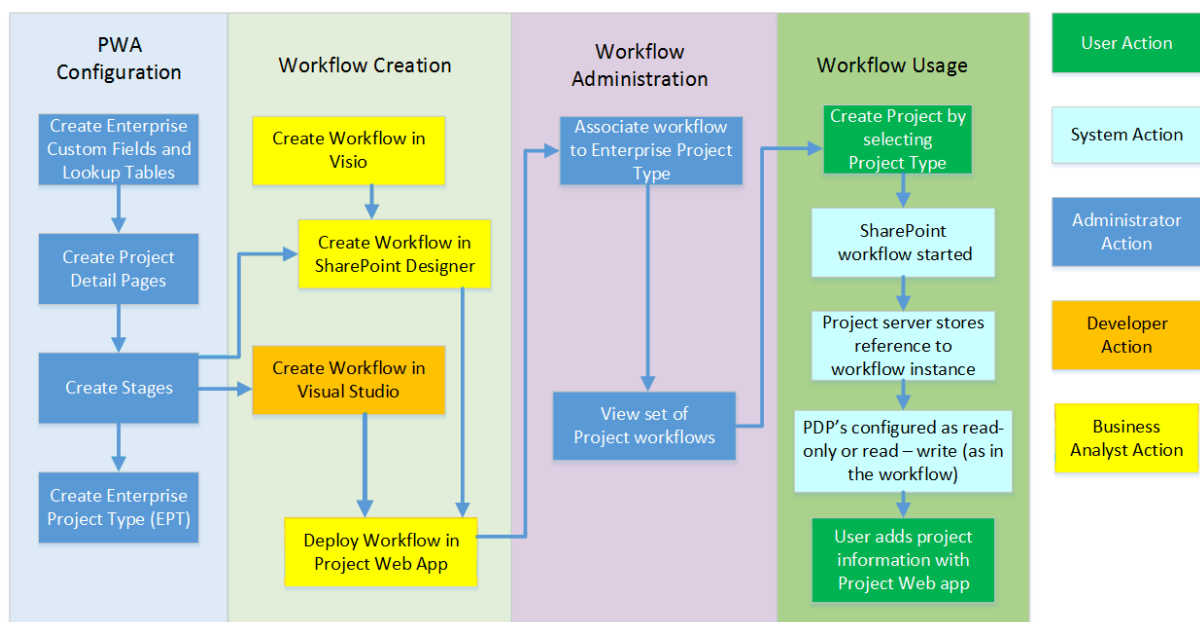


Figure 35: Different actors for managing a workflow

5.4 Relations between Project Server Objects and Workflow orchestration

Figure 36 shows the relation between the Project Server objects and the workflow objects that can be referenced from SharePoint Designer or Visual Studio.

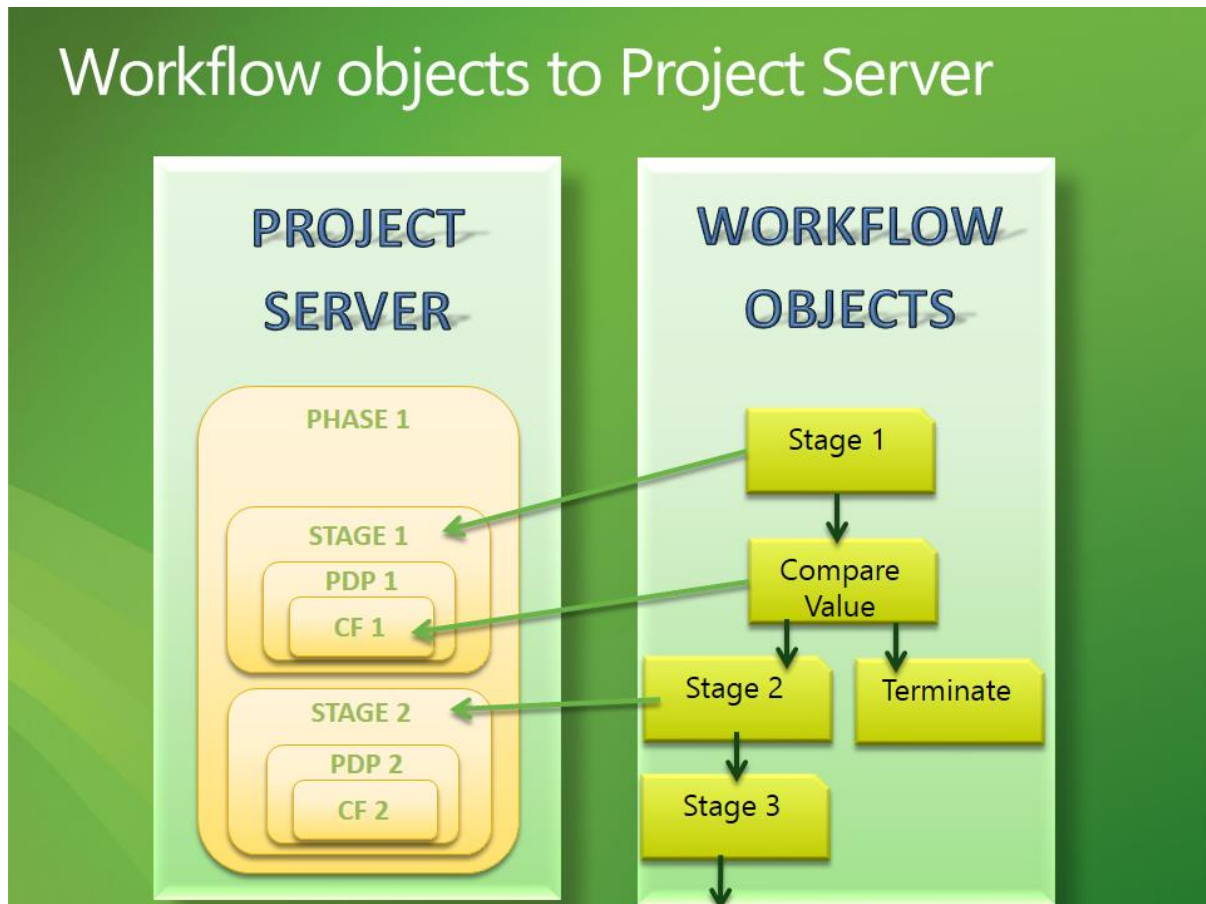


Figure 36: Relation between Project Server and Visual Studio/SPD objects

The custom fields (CF 1 and CF 2) are values that can be used in SharePoint Designer as project data to make a decision based on the value or to store the value.

The Project Detail Pages (PDPs) are Web Part pages that include custom fields.

The stages hold one or more PDPs.

The Phases group Stages together.

So from a user perspective, as the user goes from stage to stage he/she is exposed to a different set of pages that expose different types of information.

The workflow determines which stages appear, and in what order, based on the sequence that you define with SharePoint Designer.

The example in Figure 36 shows that when you create the workflow in Visual Studio, you reference the following objects that reside within Project Server:

- The activity at the top points to Stage 1, which tells the workflow that Stage 1 should be displayed first.

- The Compare Value activity performs If/Else logic based on a value in a custom field. Depending on the comparison result, additional activities point to different stages or terminate the workflow.
- The key point is that you do not set the stage order in Project Server. Project Server has no way to determine in what order the stages should be. The ordering is completely dictated by the workflow, and the orchestration of the workflow is controlled by calculations or logic within the workflow.

5.5 Decision tree between SharePoint Designer and Visual Studio

Among the greatest advantages of the workflow framework in SharePoint 2013 is the ease with which information workers can use the no-code environment of SharePoint Designer to create rich and powerful workflows. Additionally, a high degree of flexibility and customization is available in a declarative authoring environment such as Visual Studio.

Both of these workflow authoring environments—SharePoint Designer and Visual Studio—offer specific advantages and disadvantages. In this section, we explore how to determine which authoring environment best suits your workflow development needs.

Using SharePoint Designer

- Target users: Information workers, business analysts, SharePoint developers.
- Difficulty level: Familiarity with SharePoint Designer, including the core workflow components, such as stages, gates, actions, conditions, and loops.

With SharePoint Designer, users can create a workflow that is attached to a list, library, or site using a no-code, text-based designer. Or, they can use the new visual design environment in which graphical elements are arranged on a design surface to represent the logical flow of a business process. SharePoint Designer excels at enabling rapid workflow development by non-technical workers.

Using Visual Studio

- Target users: Intermediate or advanced software developers.
- Difficulty level: Familiarity with Visual Studio, including software development concepts such as event receivers, packaging and deployment, and security.

5.5.1 Comparing SharePoint Designer workflows versus Visual Studio Workflow

	SharePoint Designer	Visual Studio
Reusability	Create reusable WF	Create WF templates
Include in SharePoint apps	Yes	Yes
Custom Code	No	No
Custom Actions	Consume, not create	Yes; underlying activities
Visio Integration	Yes	No
Debugging	No	Yes

Figure 37: Comparing SharePoint Designer and Visual Studio for implementing Workflows

The two primary tools that are used to create custom workflows in SharePoint 2013, SharePoint Designer and Visual Studio, are similar but have some distinct differences from their previous versions:

- **Reusability:** Similar to previous versions, SharePoint Designer can create reusable workflows that can be used multiple times within the same site or exported for deployment to other sites. Visual Studio workflows are created as templates which can be deployed and associated with any list or library.
- **Deployment:** Workflows authored with SharePoint Designer are automatically added to the site or list at design time. These types of workflows cannot be included in an app package using the new SharePoint app model. Visual Studio based workflows are written as templates that can be included in a new SharePoint app and associated with a list or library.
- **Custom Code:** Neither Visual Studio nor SharePoint Designer can create workflows with custom code. This is a big change from SharePoint 2010 where Visual Studio workflows were based on custom code. In SharePoint 2013, all workflows are declarative. When you need to use custom code to implement logic in a workflow, you should use either the new activities for calling SOAP/REST Web services (available to SharePoint Designer and Visual Studio) and include the logic in a web service or create a custom activity/action which can be used in either tool.



- **Custom Actions:** Both SharePoint Designer and Visual Studio can consume and use custom actions in authored workflows. However Visual Studio will use the underlying action's activity and not the action in its workflow. Further custom activities/actions can be created using Visual Studio.

Authoring workflows in Visual Studio provides flexibility to create workflows that support virtually any business process, regardless of its complexity, and enables debugging and reuse of workflow definitions. Perhaps most important, Visual Studio lets developers include SharePoint workflows as part of a broader SharePoint solution or app for SharePoint.

Visual Studio enables developers to create custom actions for consumption by SharePoint Designer, and provides the means to execute custom logic. With Visual Studio, developers can also create workflow templates, which can be deployed to multiple sites.

The following table provides a side-by-side comparison of the features and requirements for using SharePoint Designer and Visual Studio to create SharePoint workflows

Feature / Requirement	SharePoint Designer	Visual Studio
Enables rapid workflow development	Yes	Yes
Enables reuse of workflows	A workflow can be used only by the list or library on which it was developed. However, SharePoint Designer provides reusable workflows that can be used multiple times within the same site.	A workflow can be written as a template so that after it is deployed, it can be reused and associated with any list or library.
Include a workflow as part of a SharePoint solution or app for SharePoint	No	Yes

Enables you to create custom actions	No. However, SharePoint Designer can consume and implement custom actions that are created and deployed by using Visual Studio.	Yes. However, be aware that in Visual Studio, the underlying activities, not their corresponding actions, are used.
Enables you to write custom code	No	<p> Note</p> <p>This is changed from previous versions. In SharePoint 2013, workflows are declarative only and Visual Studio relies on the visual design surface for workflow development.</p>
Can use Visio Professional to create workflow logic	Yes	No
Deployment	Deployed automatically to list, library, or site on which they were created.	Create a SharePoint solution package (.wsp) file and deploy the solution package to the site (SPWeb).
One-click publishing available for workflows	Yes	Yes
Workflows can be packaged and deployed to a remote server	Yes	Yes
Debugging	Cannot be debugged.	<p>Workflow can be debugged by using Visual Studio.</p> <p> Important</p> <p>This feature is not implemented in the current build.</p>

Can use only actions that are approved by site administrator	Yes	<p>Yes</p> <p>Note</p> <p>This is changed from previous versions. Previously, workflows and actions that were authored by using Visual Studio were code-based and deployed at the farm scope, so administrator approval was not required.</p>
--	-----	--

Figure 38: Features and Requirements comparison between SharePoint Designer and Visual Studio

5.6 Using SharePoint Designer 2013 to import Visio Diagrams

This chapter shows how we can import the workflow designed in Visio 2013 using SharePoint Designer 2013.

Open SharePoint Designer 2013 and connect to a Project Web App instance (on-premises or Project Online).

Select the **Workflows** object in the Site Objects Panel on the left.

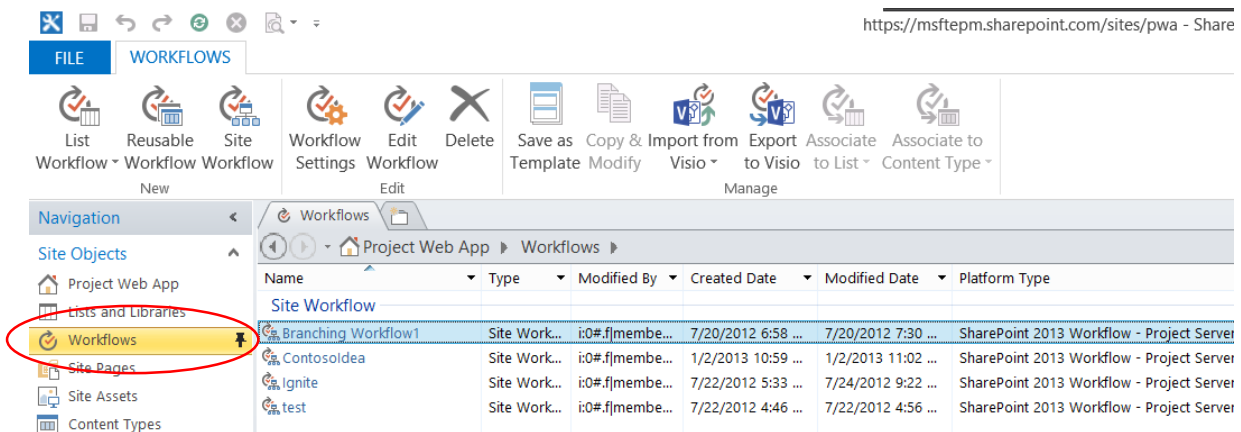


Figure 39: Using SharePoint Designer 2013

In the toolbar, select **Import from Visio**, and then in the drop-down list, select **Import Visio 2013 Diagram**.



Figure 40: Importing the workflow from Visio 2013

A dialog box opens to create a new workflow

In **Workflow Type**, select **Project Workflow**.

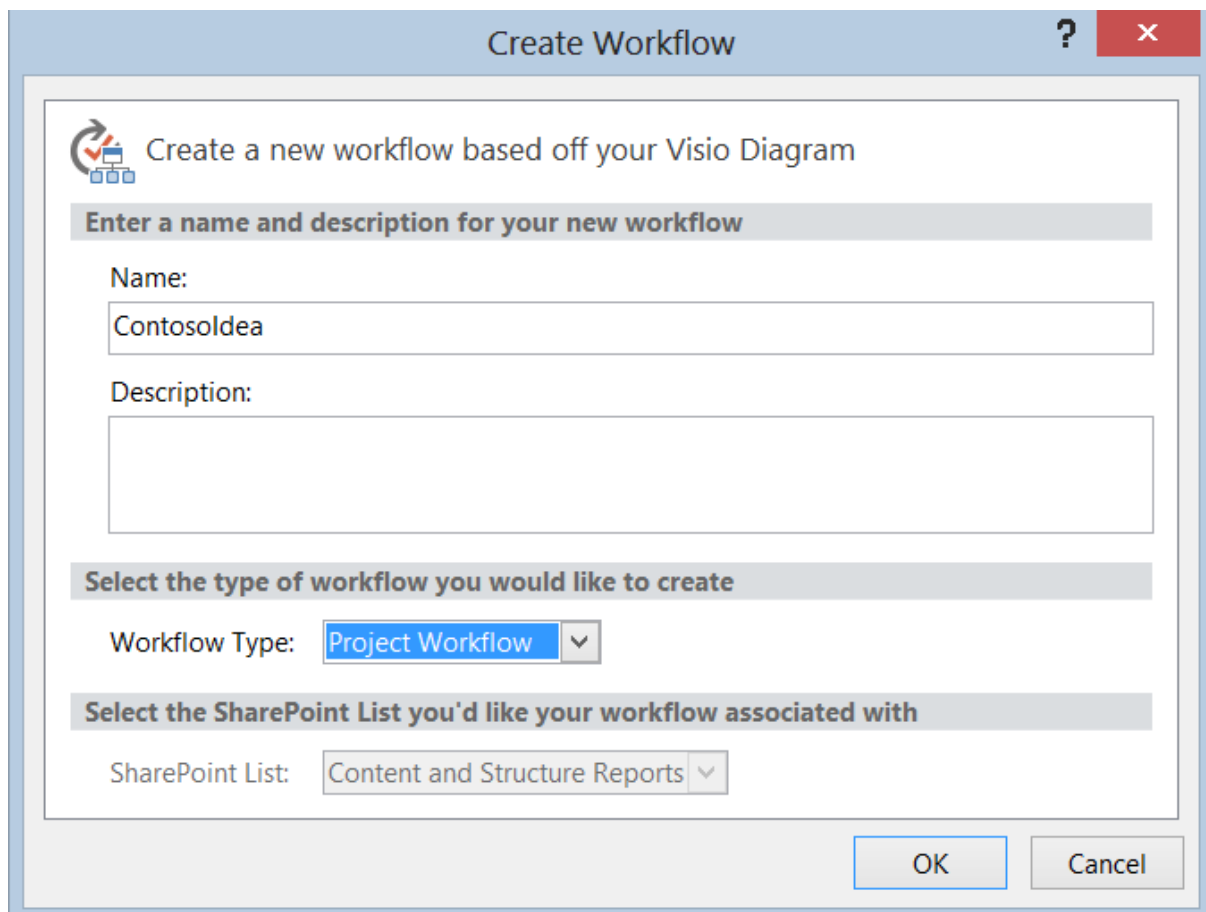


Figure 41: Creating a workflow from a Visio 2013 import

SharePoint Designer opens a new Visual Designer window.

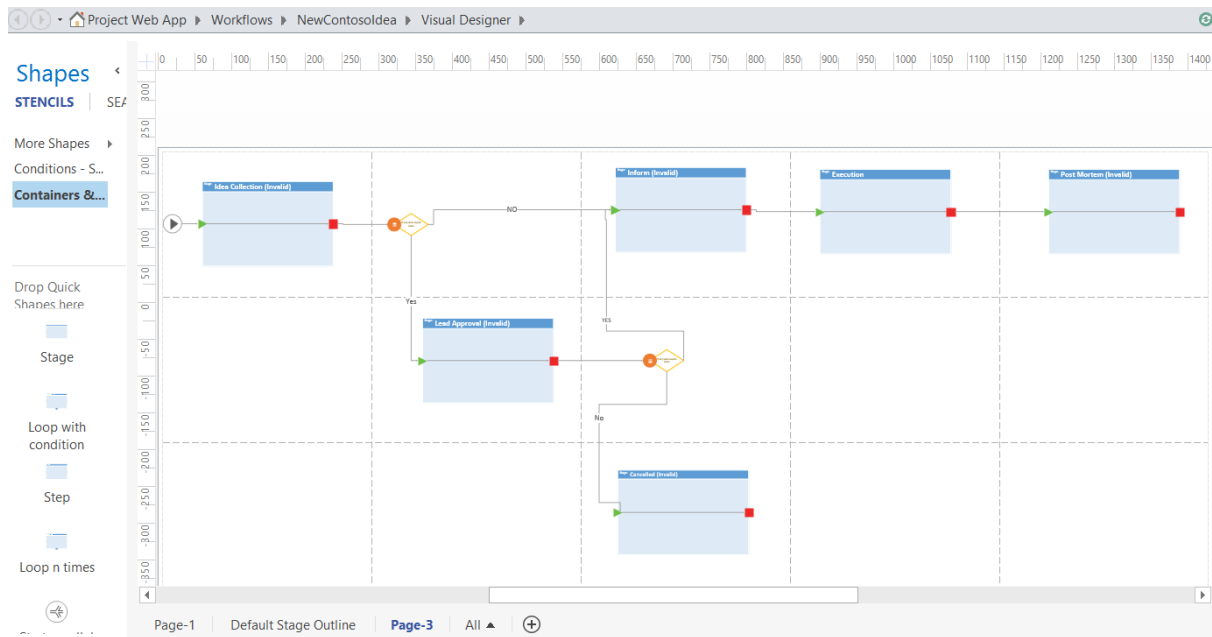


Figure 42: Result of Visio 2013 import

We can now use SharePoint Designer 2013 to implement the workflow (See chapter 5.7.2).

5.7 Using SharePoint Designer 2013

This chapter shows how we use SharePoint Designer 2013 to implement the workflow.

Open SharePoint Designer 2013 and connect to a Project Web App instance (on-premises or Project Online)

We can either:

- Import a Visio diagram (described in the previous chapter), or
- Create a new site workflow

5.7.1 Using a new site workflow

Create a new workflow by choosing **Site Workflow** in the ribbon:

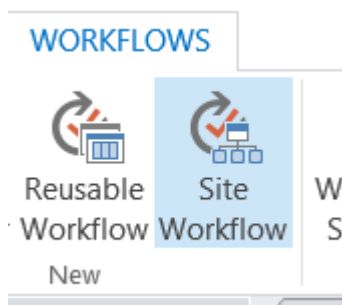


Figure 43: Creating a site workflow in SharePoint Designer 2013

For **Platform Type**, select **SharePoint 2013 Workflow – Project Server**.

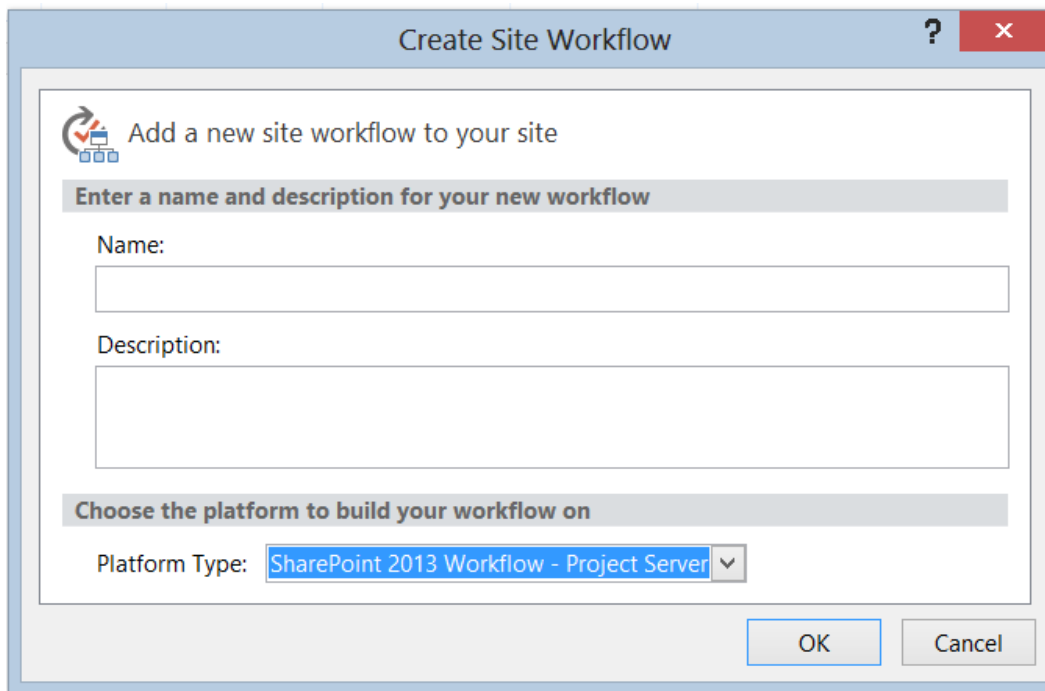


Figure 44: SharePoint Site Workflow Dialog box

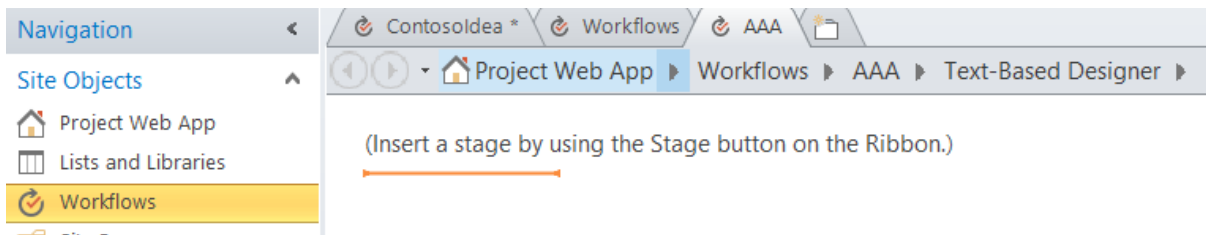


Figure 45: New workflow using Text-Based Designer

When we are connected to the PWA instance, we can select the existing stage that is defined in chapter 4.7.4.

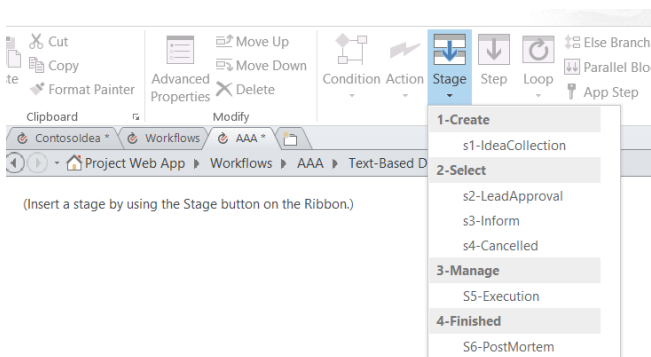


Figure 46: Using the PWA defined stages from SharePoint Designer 2013

5.7.2 Using the Visio import

After importing the Visio diagram (described in chapter 5.6), select the **Workflows** object in the Site Objects Panel on the right.

From SharePoint Designer 2013 select the **Default Stage Outline** view:

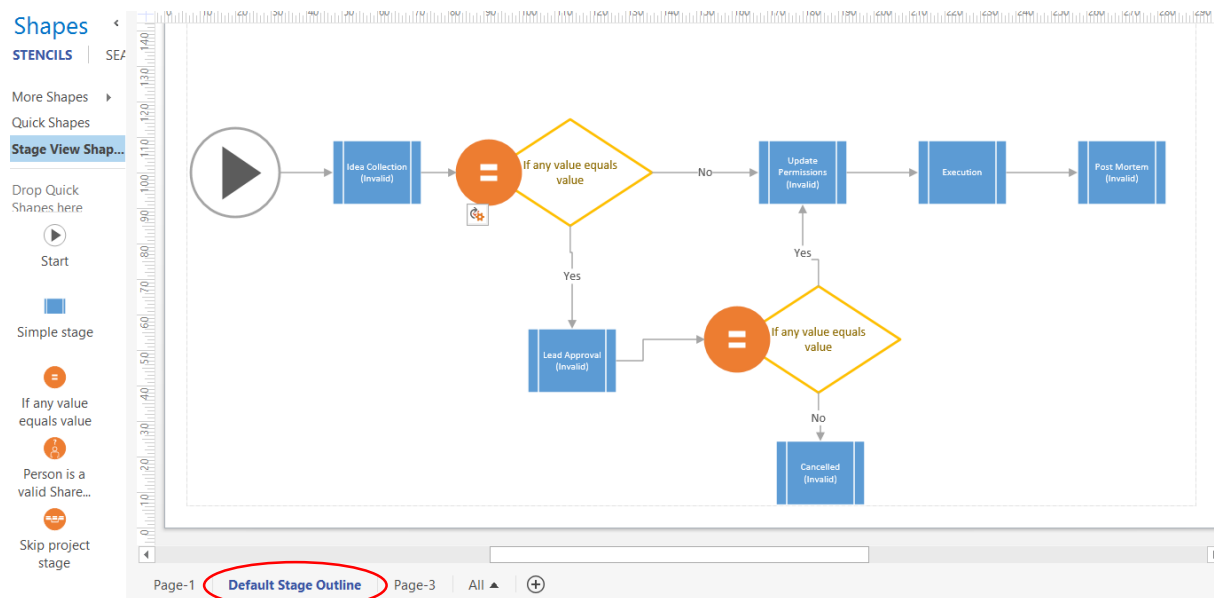


Figure 47: Default Stage Outline

Because the stages in the Visio diagram use different names than the objects defined in Project Server, the stages must be mapped manually; otherwise, automatic mapping would be incorrect (Figure 48 shows the **Invalid** keyword).

Select the **Stage** property

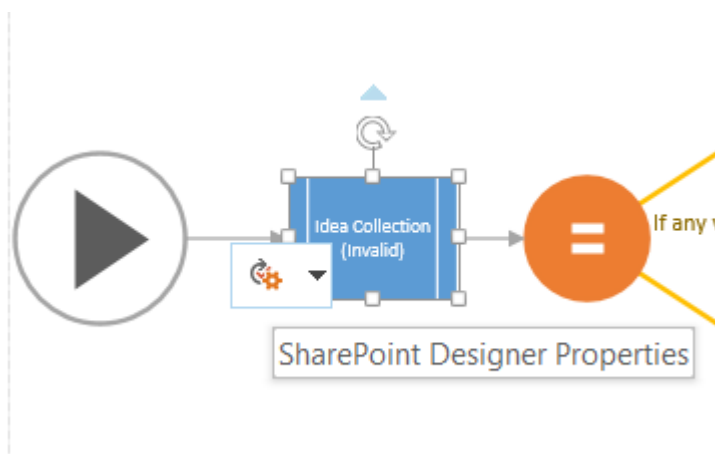


Figure 48: SharePoint Designer Properties

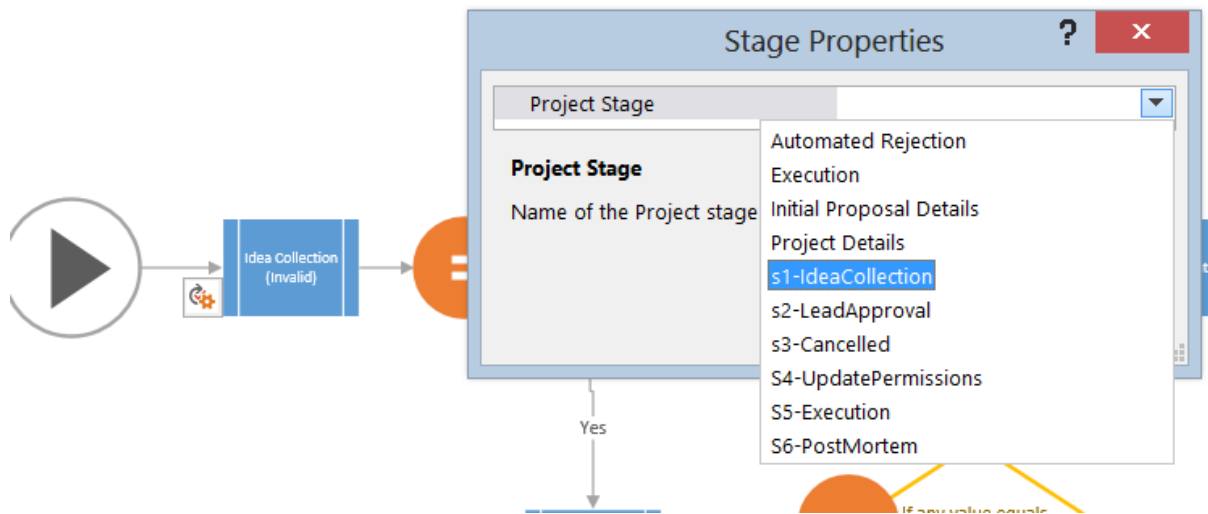


Figure 49: Selecting the right PWA stage from SharePoint Designer

Select the appropriate stage from the proposed list

Refresh the stages that may have been defined after the site was opened in SharePoint Designer.

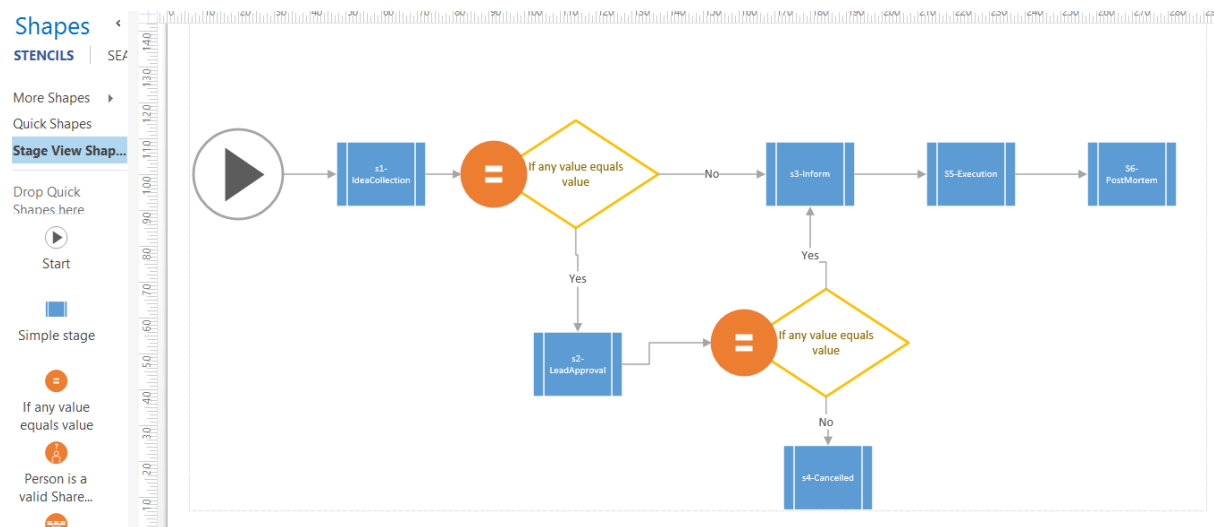


Figure 50: Workflow in SharePoint Designer after remapping the correct PWA stage

Check for any errors by choosing **Check for Errors**.

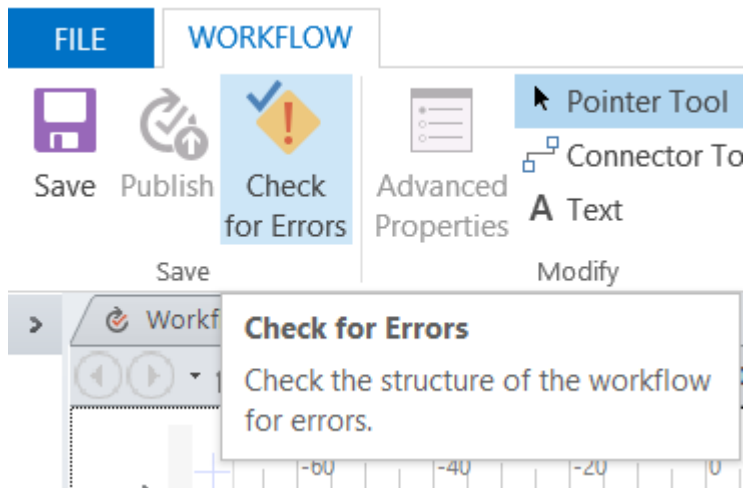


Figure 51: Check for errors in SharePoint Designer 2013

If there are no errors,

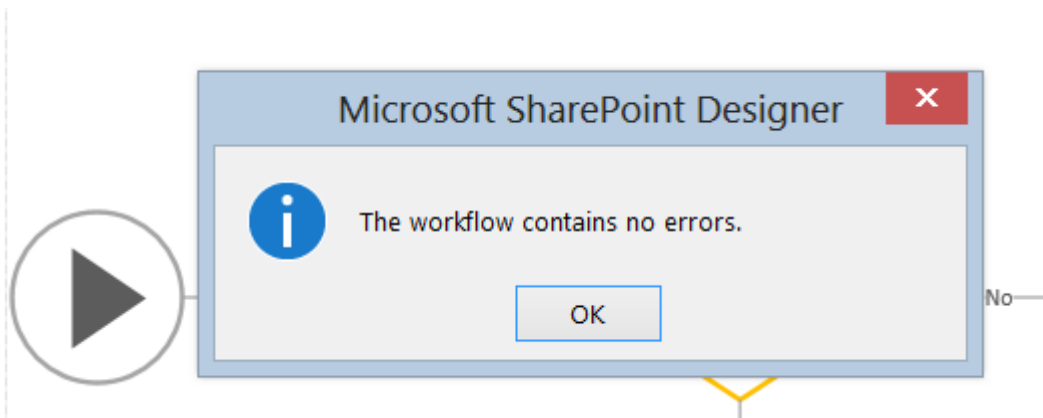


Figure 52: No errors in SharePoint Designer

you can choose **Generate Workflow Outline**.

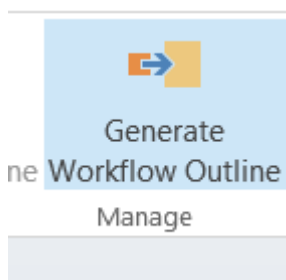


Figure 53: Generate Workflow outline view

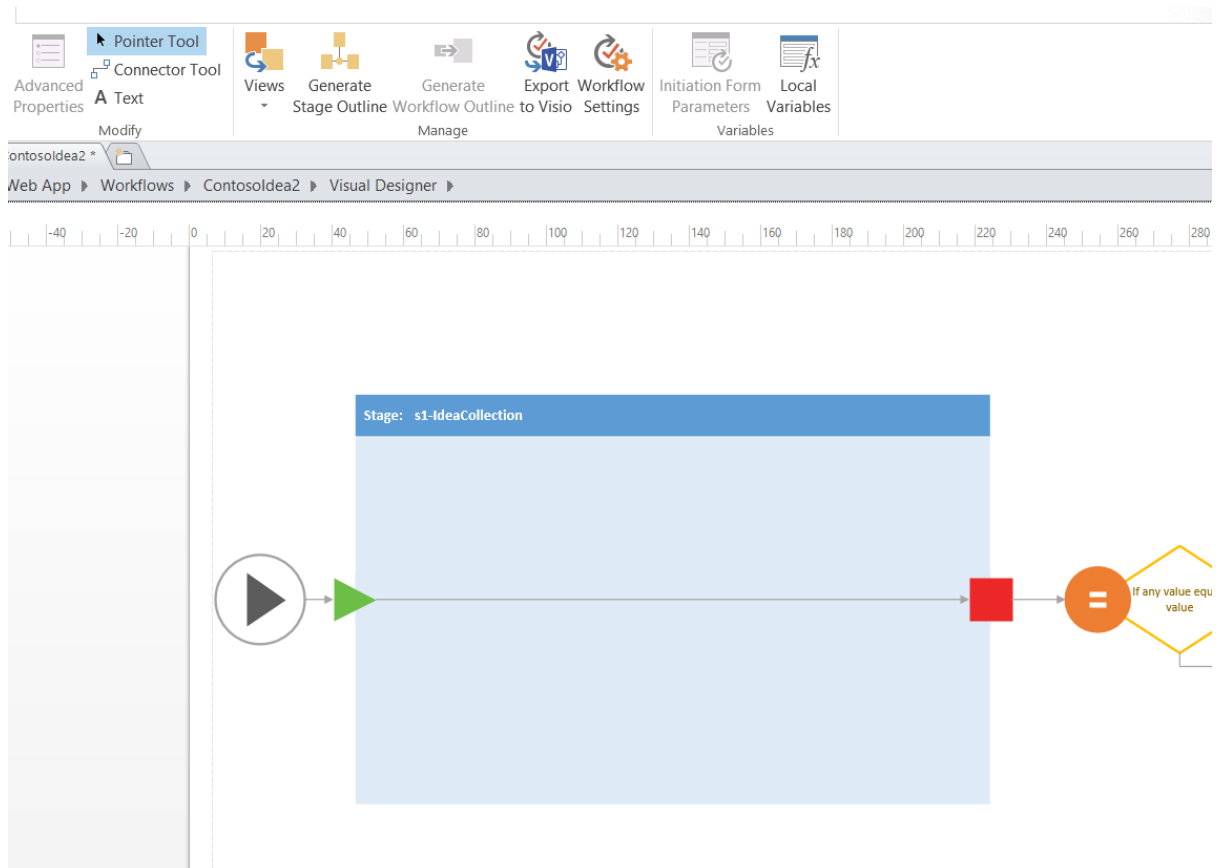


Figure 54: Result of Workflow outline

To continue the workflow implementation, switch to the **Text-Based Designer** view.

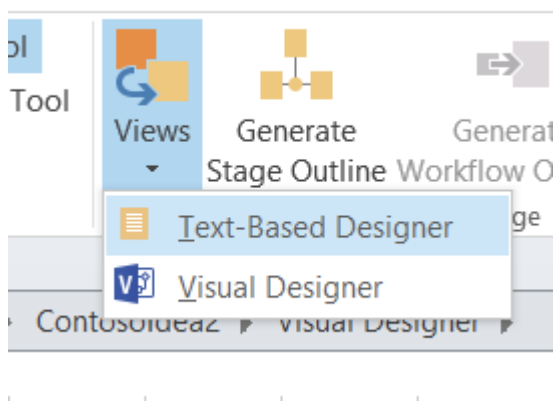


Figure 55: Switching to Text-Based Designer

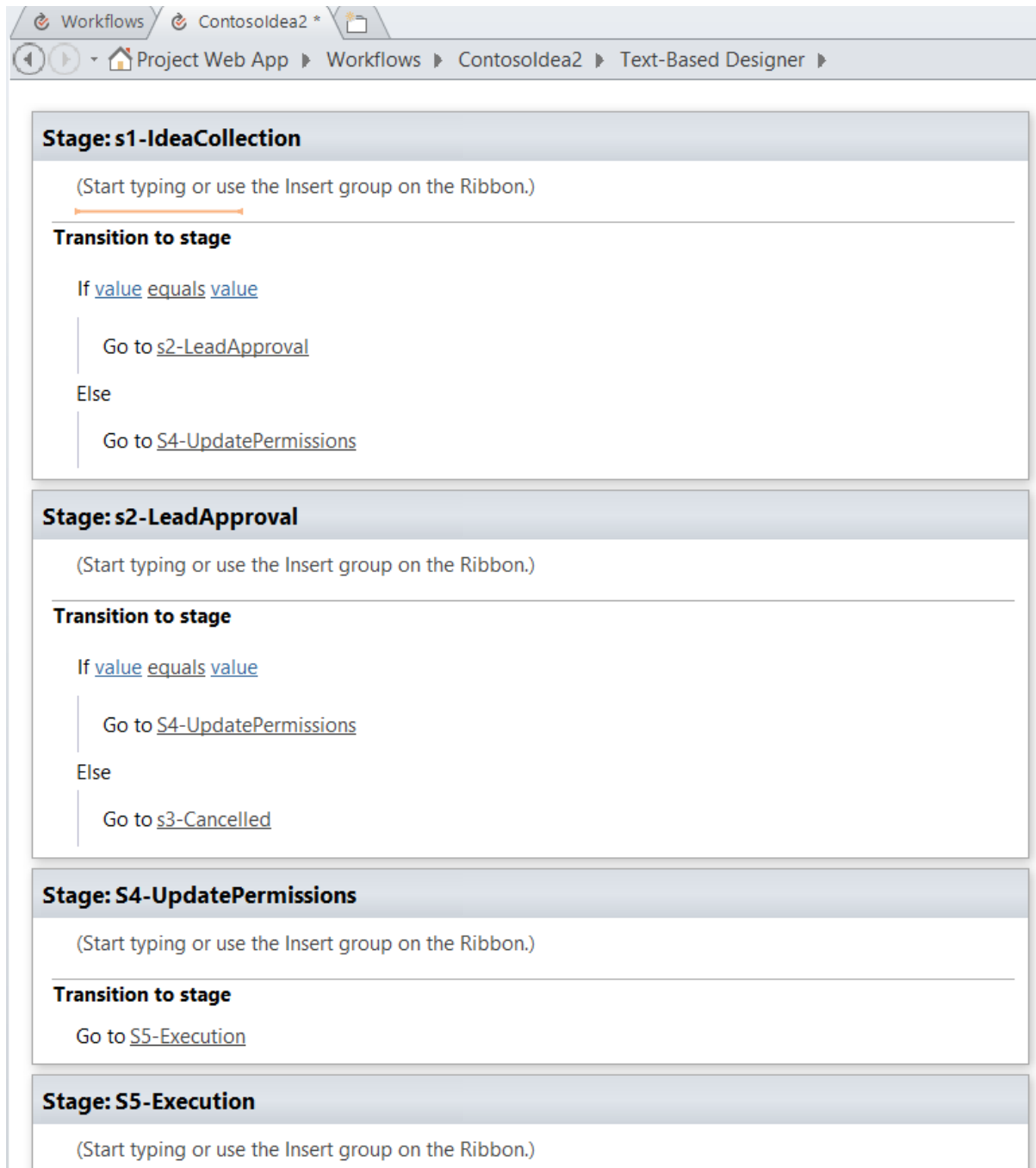
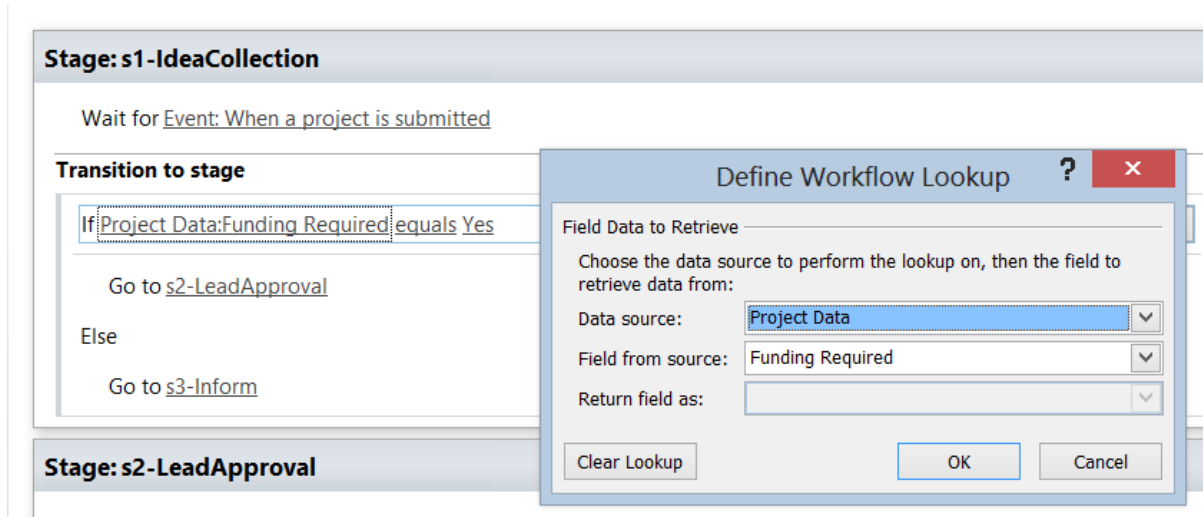


Figure 56: Initial Workflow in Text-Based Designer

The stage s1 includes the definition of the test for funding, where we test the value of the **Funding Required** enterprise custom field.



Transition to stage

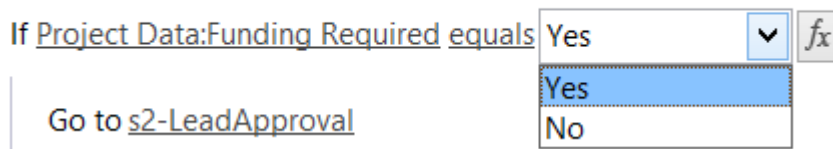


Figure 57: Testing if a Funding is required from SharePoint Designer 2013

In stage s3, we send an email.

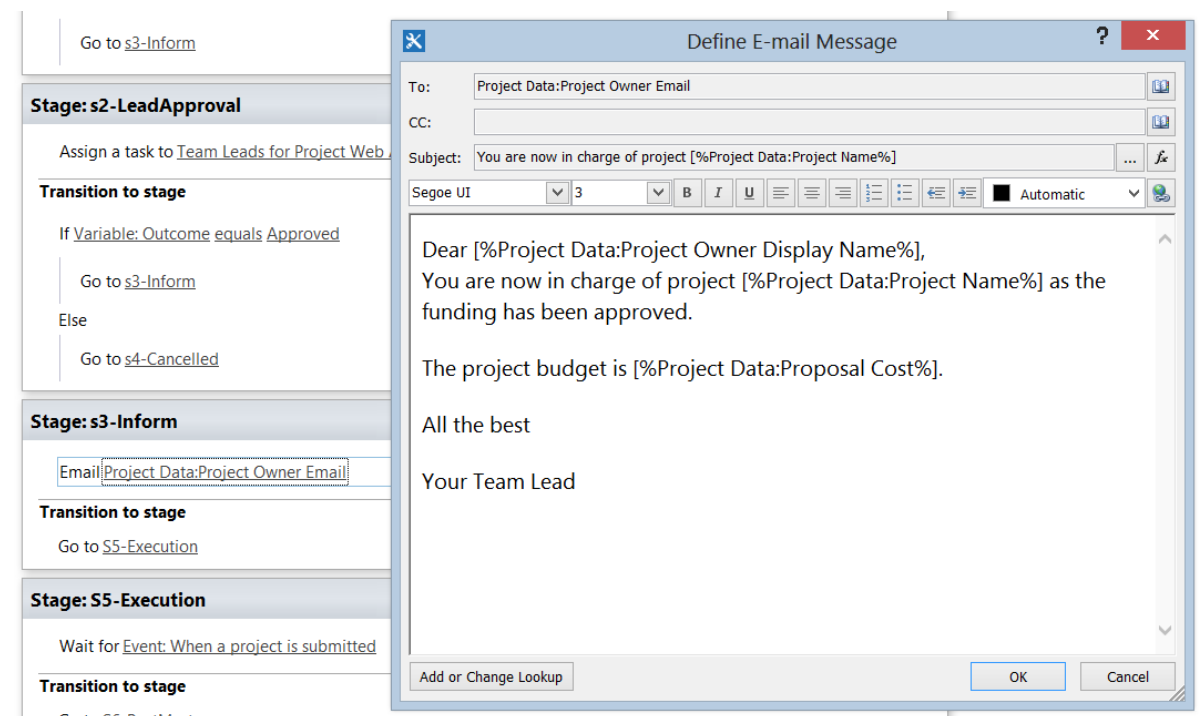


Figure 58: Using a sending email Action

We use a variable for the email that enables us to have dynamic text that is generated.

Figure 59 shows the resulting email.

You are now in charge of project test 00

✕ DELETE ← REPLY ⇐ REPLY ALL → FORWARD ...



workflow@noreply
Tue 1/22/2013 4:39 PM

mark as unread

To: ☐ Steven Haden PPM;

Dear Steven Haden PPM,
You are now in charge of project test 00 as the funding has been approved.

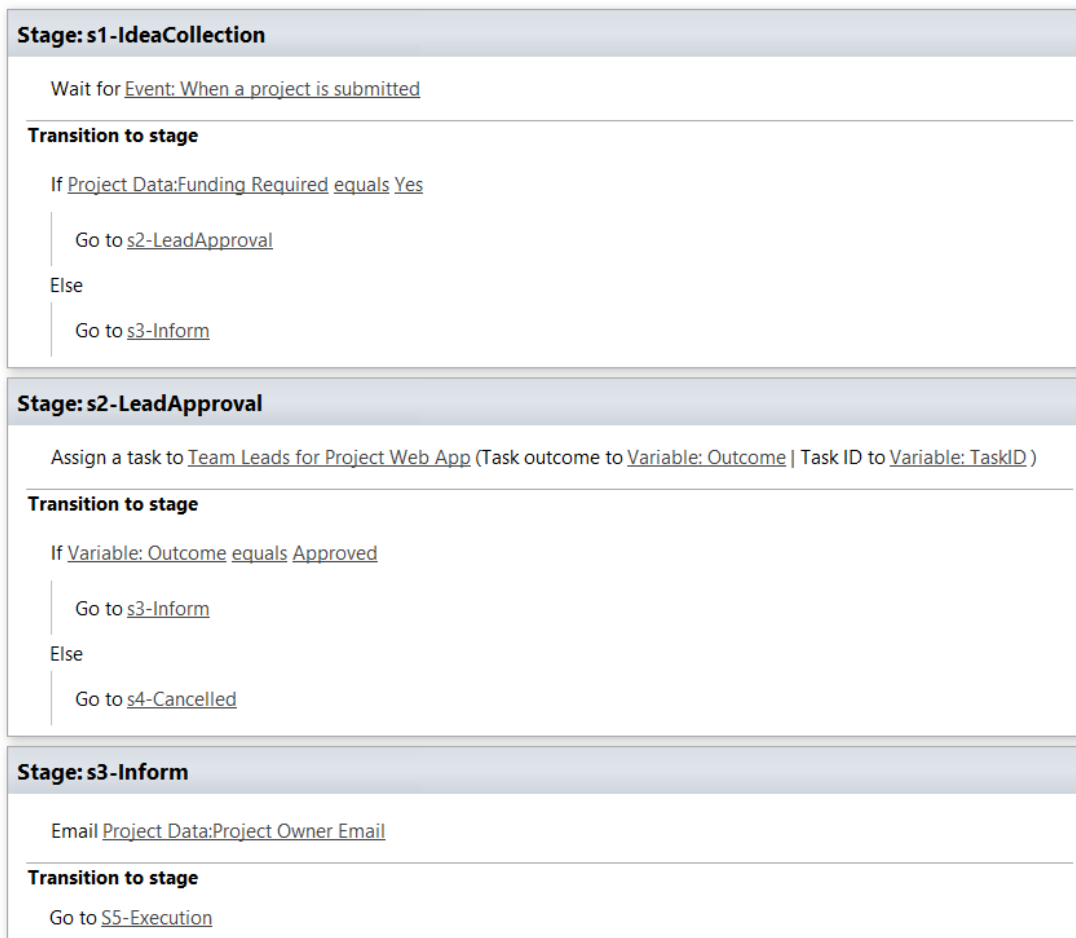
The project budget is \$5,000.00.

All the best

Your Team Lead

Figure 59: Generated email when the project is approved

After updating the workflow with the different actions, Figure 60 shows the final result:



Stage: S5-Execution Wait for <u>Event: When a project is submitted</u> <hr/> Transition to stage Go to <u>S6-PostMortem</u>
Stage: S6-PostMortem Wait for <u>Event: When a project is submitted</u> <hr/> Transition to stage Go to <u>End of Workflow</u>
Stage: s4-Cancelled (Start typing or use the Insert group on the Ribbon.) <hr/> Transition to stage Go to <u>End of Workflow</u>

Figure 60: Workflow in Text-Based Designer after updates

5.8 Using Visual Studio 2012 to create Project Workflows

For information about creating SharePoint workflows with Visual Studio, see [http://msdn.microsoft.com/en-us/library/jj163199\(v=office.15\).aspx](http://msdn.microsoft.com/en-us/library/jj163199(v=office.15).aspx).

For a specific example for Project Server, see http://blogs.msdn.com/b/project_programmability/archive/2012/11/07/creating-project-workflows-using-visual-studio-2012.aspx.

Unlike earlier versions, workflows in SharePoint 2013 are entirely declarative. Built on Windows Workflow Foundation 4, Visual Studio provides a visual workflow designer surface that lets you create the following entirely in the designer environment:

- Custom workflows
- Workflow templates
- Forms
- Custom workflow activities

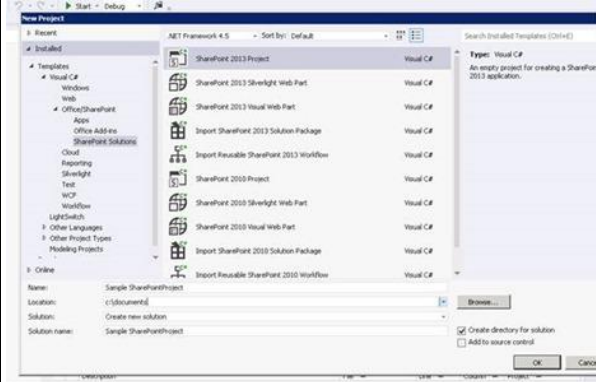

Your workflow is packaged and deployed as a SharePoint Feature.

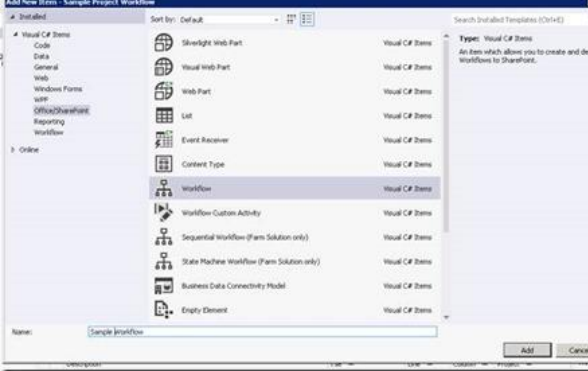
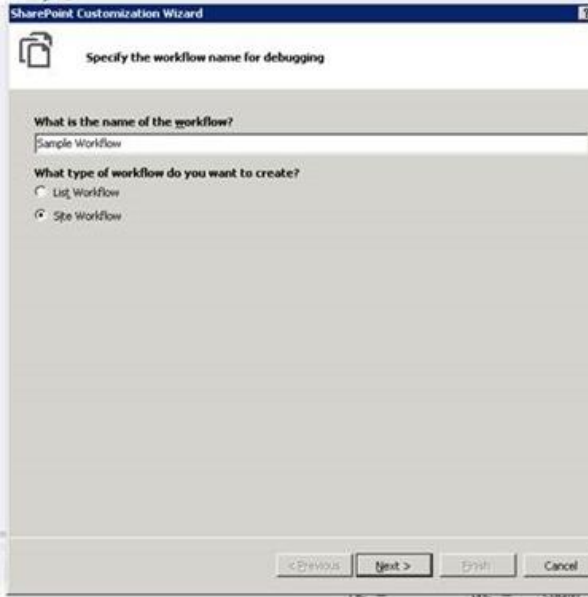
Perhaps the most significant change for Visual Studio developers is that custom workflows are no longer compiled and deployed as .NET Framework assemblies. Furthermore, SharePoint 2013 no longer uses InfoPath forms; instead, forms generation relies on ASP.NET forms.

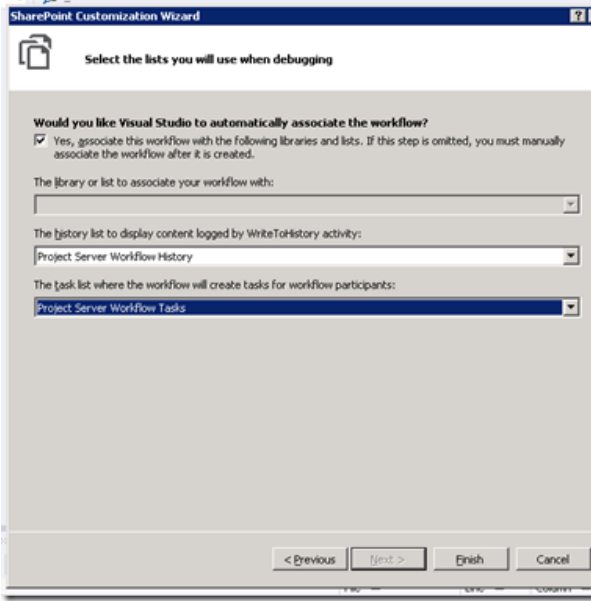
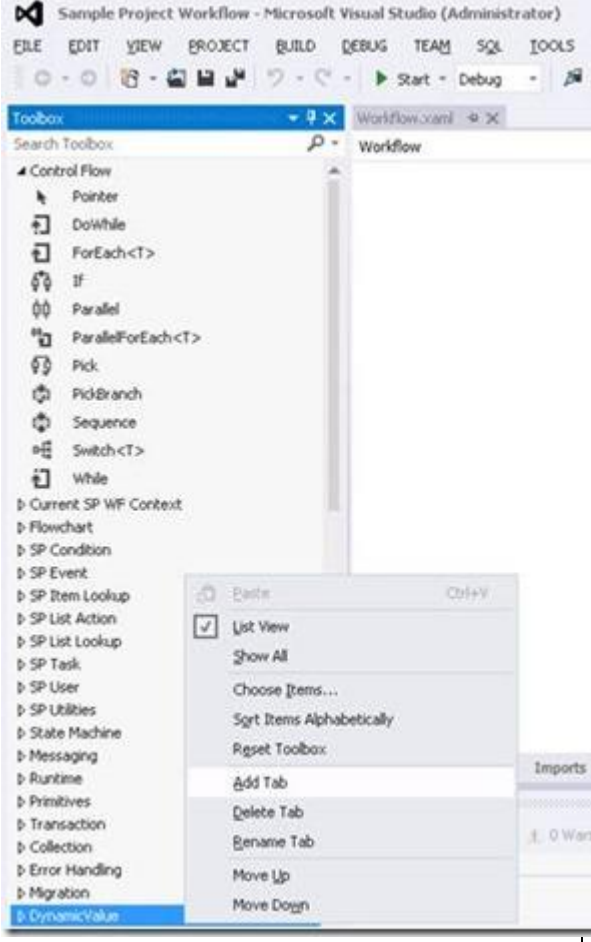
Finally, the Visual Studio workflow project templates have changed. Whereas templates for state machine and sequential workflows were previously provided, these distinctions are no longer meaningful. Rather, Visual Studio project templates are available in the Visual Studio build provided on your virtual machine (VM).

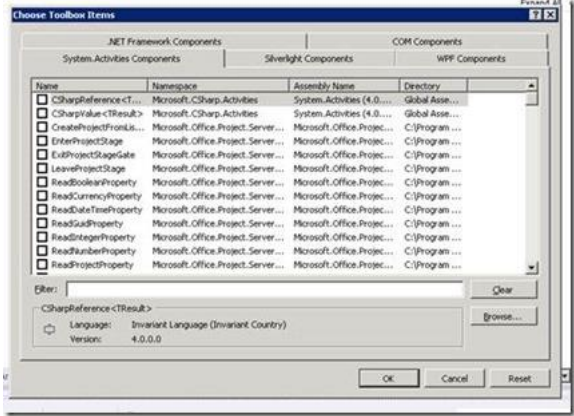
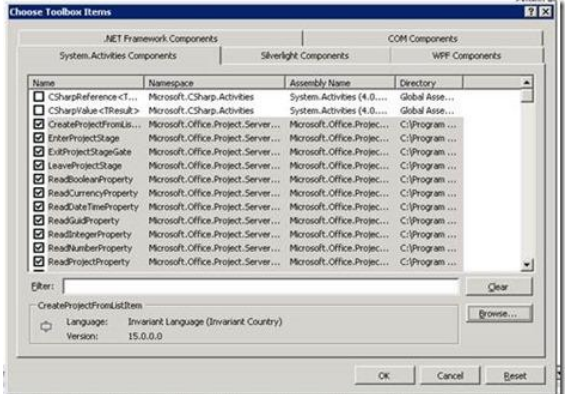
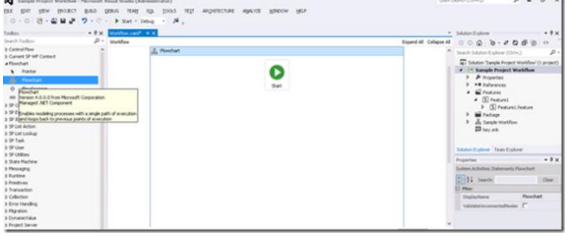
If you use Visual Studio 2012 to develop workflows, you should use the [Microsoft.ProjectServer.Client.WorkflowActivities](#) class in the client-side object model (CSOM). The CSOM supports both online applications and on-premises applications for Project Server 2013.

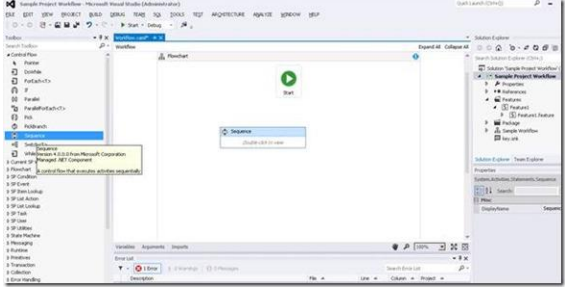

This section assumes you have access to an on-premises installation of Project Server 2013.

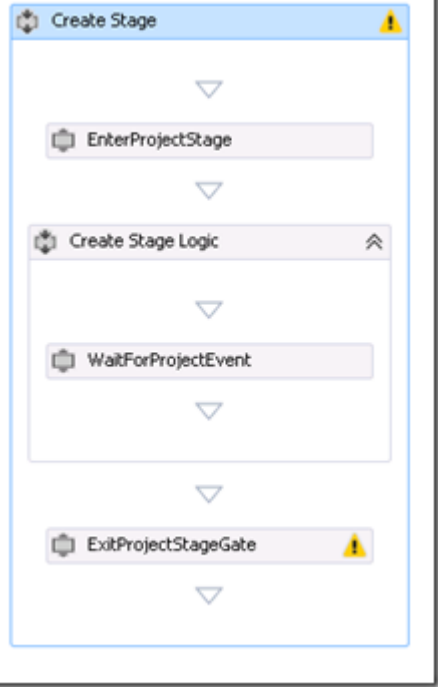
Actions	Screen
<p>Create the Workflow solution: File Menu>New>Project >Office/SharePoint>SharePoint solutions > SharePoint 2013 Project.</p> <p>Give this project a name, hit OK</p>	
<p>In the customization wizard, enter the address of the PWA web where the workflow will be published.</p> <p>Select the sandboxed solution option to limit the workflow to the specified PWA web.</p>	

Actions	Screen
<p>In the Project menu, click Add New Item, and then on the Office/SharePoint tab, select Workflow, enter a name, and choose Add.</p>	
<p>In the customization wizard, enter a name; Sample Workflow and choose Site workflow.</p>	

Actions	Screen
<p>Pick the history list and the workflow tasks list from that site.</p> <p>We recommend that you use the default lists because a number of PWA UI entry points use these default lists. Choose Finish.</p>	 <p>The screenshot shows the 'SharePoint Customization Wizard' dialog box. It has a title bar with a question mark icon. The main text says 'Select the lists you will use when debugging'. Below this, there's a section 'Would you like Visual Studio to automatically associate the workflow?' with a checked checkbox and a note: 'Yes, associate this workflow with the following libraries and lists. If this step is omitted, you must manually associate the workflow after it is created.' There are three dropdown menus: 'The library or list to associate your workflow with:', 'The history list to display content logged by WriteToHistory activity:', and 'The task list where the workflow will create tasks for workflow participants:'. The last two are set to 'Project Server Workflow History' and 'Project Server Workflow Tasks' respectively. At the bottom are buttons for '< Previous', 'Next >', 'Finish', and 'Cancel'.</p>
<p>We need to set up the environment to use the Project Server activities.</p> <p>In the toolbox, right click and choose Add Tab in the option menu. Name the new tab Project Server.</p>	 <p>The screenshot shows the Visual Studio interface. The title bar says 'Sample Project Workflow - Microsoft Visual Studio (Administrator)'. The menu bar includes FILE, EDIT, VIEW, PROJECT, BUILD, DEBUG, TEAM, SQL, and TOOLS. The toolbar has icons for file operations, debugging, and project management. The 'Toolbox' window is open on the left, showing a search bar and a list of activities under 'Control Flow'. A right-click context menu is open over the toolbox, showing options like Paste, List View, Show All, Choose Items..., Sort Items Alphabetically, Reset Toolbox, Add Tab, Delete Tab, Rename Tab, Move Up, and Move Down. The 'Add Tab' option is highlighted.</p>

Actions	Screen
<p>Right click the Project Server tab and select Choose items in the option menu to show the Choose Toolbox Items dialog box.</p>	
<p>In the dialog box, choose Browse, and navigate to where the workflow DLLs are located. They are usually located in C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\15\TEMPLATE\WorkflowActivities. You'll see two activities assemblies there. Open the project server assembly (Microsoft.Office.Project.Server.WorkflowActivities.dll), and choose OK. You are now taken back to the Toolbox Items dialog box, which highlights the selected corresponding activities. Choose OK to continue.</p>	
<p>You might see a "sequence" on the workflow designer canvas. Delete that, and from the toolbox, pick Flowchart and add it by dragging it into the main area. This flowchart will be the main container of all the stages of the workflow.</p>	

Actions	Screen
<p>In the toolbox, click Control flow, and add the sequence inside that flowchart. Throughout the sample workflow, we will use the sequence to represent workflow stages in Visual Studio. This is similar to how SharePoint Designer handles each stage. Each stage is equivalent to a separate sequence in Visual Studio.</p>	 <p>The screenshot shows the Visual Studio Workflow Designer interface. On the left, the 'Toolbox' pane is open, and the 'Control flow' category is selected. A 'Sequence' activity is being added to the flowchart. The main canvas shows a 'Start' activity (green circle with a play button) and a 'Sequence' activity (blue rectangle) connected by a line. The right-hand side shows the 'Properties' pane for the 'Sequence' activity.</p>
<p>Rename the sequence to Create Stage by selecting the Sequence; start typing to change the name. Drag a line from Start to Create Stage to connect them together.</p>	 <p>The screenshot shows the Visual Studio Workflow Designer interface. The 'Sequence' activity has been renamed to 'Create Stage'. A line is drawn from the 'Start' activity (green circle with a play button) to the 'Create Stage' activity (blue rectangle). The right-hand side shows the 'Properties' pane for the 'Create Stage' activity.</p>

Actions	Screen
<p>Double-click Create Stage to drill into the sequence.</p> <ol style="list-style-type: none"> Under Project Server in the toolbox, add the EnterProjectStage and ExitProjectStageGate activities to the sequence. These two activities are required in any of the PWA stages in Visual Studio. In the properties of EnterProjectStage, change the StageID to the Stage ID of the particular stage this sequence represents. You can find the stage ID in the URL of that stage, which is available if you navigate to that stage in PWA Settings > Workflow Stages, and then click on the particular stage. Since stageID is a string, the ID should be provided in quotation marks. Add another sequence between EnterProjectStage and ExitProjectStageGate. Essentially, the text-based designer in the SharePoint Designer stage definition represents everything in this sequence. From the Project Server section in the toolbox, drag and drop the WaitForProjectEvent activity in the Create Stage Logic sequence. Change the EventName property to OnProjectSubmit. The other supported event names are OnProjectCommit and OnProjectCheckIn. 	

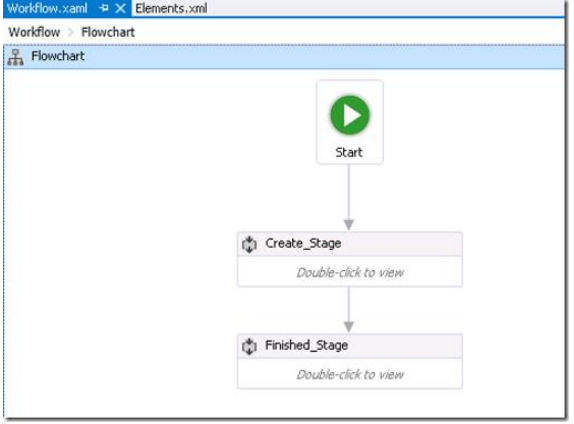
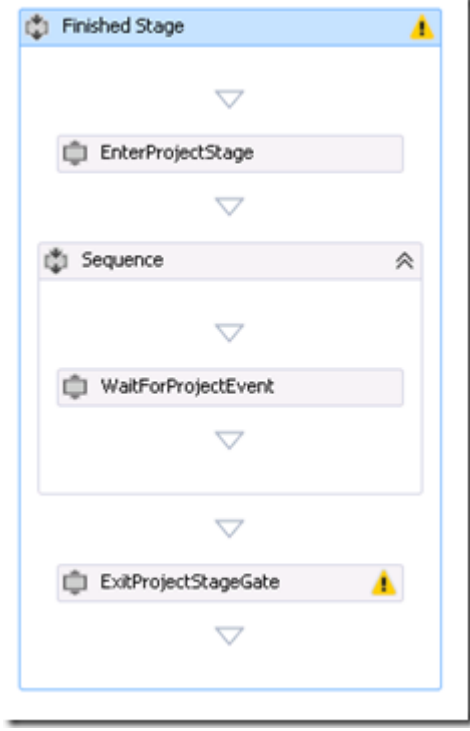
Actions	Screen
<p>In the breadcrumb, click Flowchart to go one level up. Add another sequence after Create_Stage, name it Finished_Stage, and then drag a connection from Create_Stage to Finished_Stage.</p>	
<p>Similarly, add the EnterProjectStage and ExitProjectStageGate activities to the Finished Stage sequence, and then add the WaitForProjectEvent activity and set the properties.</p>	

Figure 61: Steps to create a Project Server 2013 workflow with Visual Studio 2012

This completes building the workflow in Visual Studio. However, to ensure that the workflow can be properly published to PWA, we need to make a few more changes in the xaml files of the project. In Solution Explorer, select Elements.xaml under the workflow node.

1. Replace the **WSEventSourceGUID** with the following so that the workflow is correctly identifies as a project workflow:

```
<Property Name="WSEventSourceGUID" Value="5122D555-E672-4E5D-A7C4-8084E694A257" />
```
2. Inject the following properties under the `Url = WorkflowStartAssociation` line:

```

<Property Name="Microsoft.ProjectServer.ActivationProperties.ProjectId" Value="" />
<Property Name="Microsoft.ProjectServer.ActivationProperties.CurrentStageId" Value="" />
<Property Name="Microsoft.ProjectServer.ActivationProperties.RequestedStageId" Value="" />
<Property Name="WSEventContextKeys"
Value="Microsoft.ProjectServer.ActivationProperties.CurrentStageId;#Instance&#xA;Microsoft.Pro
jectServer.ActivationProperties.ProjectId;#Instance&#xA;Microsoft.ProjectServer.ActivationProp
erties.RequestedStageId;#Instance&#xA;" />

```

Now that everything is set, and the workflow is ready for publishing, in the Build menu, click **Build Solution**, and then click **Deploy Solution**. Visual Studio deploys the wsp file to the site.

You can also find a copy of the wsp file in the file system, in the [project name]\bin\debug directory.

The workflow is now available in PWA. If you navigate to **PWA Settings > Enterprise Project Types**, and create a new EPT, you will see the Sample workflow as one of the options in the **Workflow** drop-down list.

For more information, see [Getting started developing Project Server 2013 workflows](#) in the Project 2013 SDK.

5.9 Deploy the workflow (using SharePoint Designer)

The following steps are required to use the workflow in Project Server.

1. Publish the workflow by using SharePoint designer 2013.
2. Associate the workflow with an enterprise project type (EPT).

To publish your workflow to PWA from SharePoint Designer 2013, choose **Publish**:

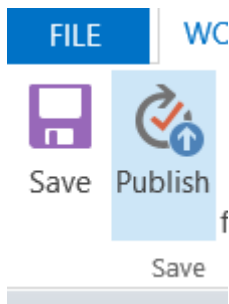


Figure 62: Publishing the workflow to PWA from SharePoint Designer 2013

When the workflow is published, you can associate the workflow with the EPT by updating the object, as in Figure 63.

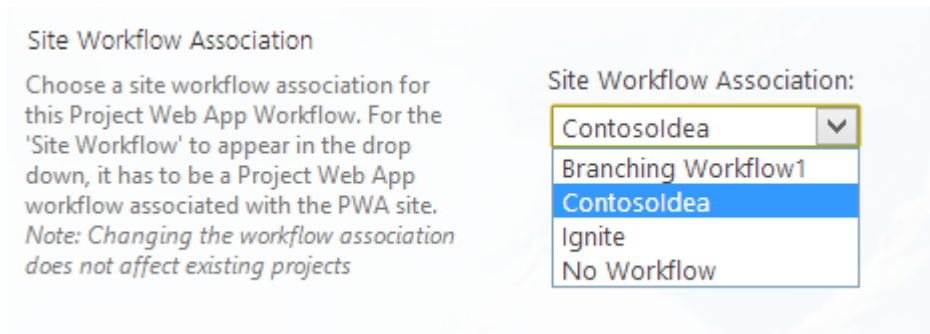


Figure 63: Associating the workflow to an EPT

You can deploy your workflow only in the PWA site that was opened with SharePoint Designer. Chapter 7 shows how a workflow can be deployed in several environments.

5.10 Understanding the Skip to Stage steps

This chapter shows how to implement steps in the *skip to stage* action and explains the result of using this action.

The skip to stage feature enables administrators to restart a workflow, where the workflow can skip through stages until it reaches a particular stage. A common scenario for this functionality is when a project has progressed to a specific point in a workflow, but needs to be pushed back to a previous stage for various reasons. For example, if a new stage was inserted that needs to be addressed, or certain fields need to be changed that are exposed only in previous stages.

Project Server workflows, like SharePoint workflows, must always be executed linearly. They cannot begin execution at random locations within a workflow. Neither can they “jump” from one point to another point, unless coded to do so within the workflow. The skip to stage functionality cannot circumvent this limitation. As such, when restarting a workflow and attempting to push it to a particular point, any activities that previously “stopped” the workflow will continue to stop it in the same place. Therefore, to fully work with the skip to stage functionality, you must wrap activities within `if` statements.

The following example illustrates the process. We will use the **Log to History List** action to understand the flow of the workflow.

In the workflow shown in Figure 64, we explicitly add conditions to test the skip to stage action: *restarts the workflow and includes this stage*.

<p>Stage: Test Create Stage 1</p> <p>Log <u>Entered Create 1</u> to the workflow history list</p> <p>If Project Web App starts the workflow normally or restarts the workflow and includes this stage</p> <p>Log <u>Waiting in Create 1</u> to the workflow history list</p> <p>then Wait for <u>Event: When a project is submitted</u></p> <p>then Log <u>Done with Create 1</u> to the workflow history list</p> <hr/> <p>Transition to stage</p> <p>Go to <u>Test Create Stage 2</u></p>
<p>Stage: Test Create Stage 2</p> <p>Log <u>Entered Create 2</u> to the workflow history list</p> <p>If Project Web App starts the workflow normally or restarts the workflow and includes this stage</p> <p>Log <u>Waiting in Create 2</u> to the workflow history list</p> <p>then Wait for <u>Event: When a project is submitted</u></p> <p>then Log <u>Done with Create 2</u> to the workflow history list</p> <hr/> <p>Transition to stage</p> <p>Go to <u>Test Execute Stage 1</u></p>
<p>Stage: Test Execute Stage 1</p> <p>Log <u>Entered Execute 1</u> to the workflow history list</p> <p>If Project Web App starts the workflow normally or restarts the workflow and includes this stage</p> <p>Log <u>Waiting in Execute 1</u> to the workflow history list</p> <p>then Wait for <u>Event: When a project is submitted</u></p> <p>then Log <u>Done with Execute 1</u> to the workflow history list</p> <hr/> <p>Transition to stage</p> <p>Go to <u>Test Execute Stage 2</u></p>
<p>Stage: Test Execute Stage 2</p> <p>Log <u>Entered Execute 2</u> to the workflow history list</p> <p>If Project Web App starts the workflow normally or restarts the workflow and includes this stage</p> <p>Log <u>Waiting in Execute 2</u> to the workflow history list</p> <p>then Wait for <u>Event: When a project is submitted</u></p> <p>then Log <u>Done with Execute 2</u> to the workflow history list</p> <hr/> <p>Transition to stage</p> <p>Go to <u>Test Finish Stage 1</u></p>
<p>Stage: Test Finish Stage 1</p> <p>Log <u>Entered Finished 1</u> to the workflow history list</p> <p>If Project Web App starts the workflow normally or restarts the workflow and includes this stage</p> <p>Log <u>Waiting in Finished 1</u> to the workflow history list</p> <p>then Wait for <u>Event: When a project is submitted</u></p> <p>then Log <u>Done with Finished 1</u> to the workflow history list</p> <hr/> <p>Transition to stage</p> <p>Go to <u>Test Finish Stage 2</u></p>
<p>Stage: Test Finish Stage 2</p> <p>Log <u>Entered Finished 2</u> to the workflow history list</p> <p>If Project Web App starts the workflow normally or restarts the workflow and includes this stage</p> <p>Log <u>Waiting in Finished 2</u> to the workflow history list</p> <p>then Wait for <u>Event: When a project is submitted</u></p> <p>then Log <u>Done with Finished 2</u> to the workflow history list</p> <hr/> <p>Transition to stage</p> <p>Go to <u>End of Workflow</u></p>

Figure 64: Example of skip to stage workflow

If we just run the workflow normally, Figure 65 shows the workflow history.



Figure 65: Result of skip to stage workflow

Note that in each stage, the workflow stops (that is, waits) and we have to click **Submit**.

When we restart the workflow, Figure 66 shows it will skip to **Test Execute Stage 2**

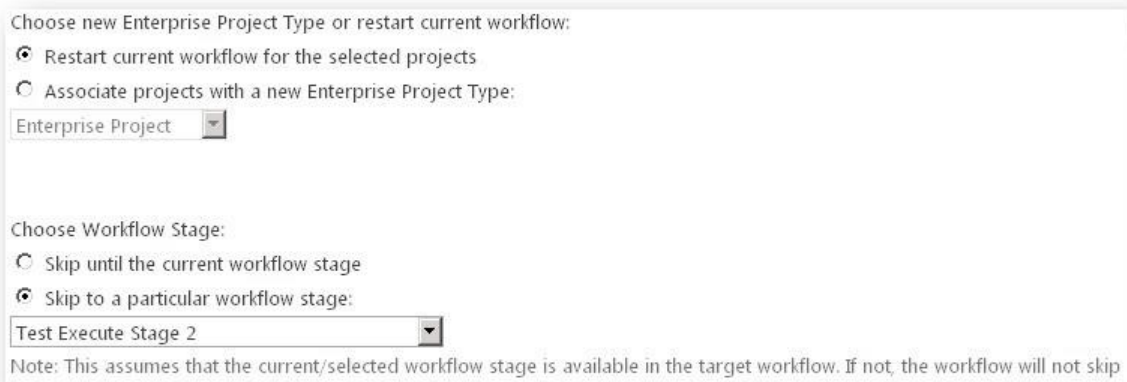


Figure 66: Restarting the workflow and skipping to stage Test Execute Stage 2

Figure 67 shows the result.



Description
Entered Create 1
Done with Create 1
Entered Create 2
Done with Create 2
Entered Execute 1
Done with Execute 1
Entered Execute 2
Waiting in Execute 2

Figure 67: Result of workflow after skipping to Stage Test Execute Stage 2

Note that the workflow still went through each stage, but it skipped around the activity that would have halted execution.

Similarly, if we manually progress the workflow through to Finished 1 (see Figure 65: Result of skip to stage workflow), then choose the **Skip until the current workflow stage** option:



Choose Workflow Stage:

☒ Skip until the current workflow stage

☐ Skip to a particular workflow stage:

Test Execute Stage 2

Note: This assumes that the current/selected workflow stage is available in the target workflow. If not, the workflow will not skip

Figure 68: Using the "Skip until the current workflow stage"

The workflow then ends in that stage, but the previous stages show that the skip path was taken (see Figure 69).



Description
Entered Create 1
Done with Create 1
Entered Create 2
Done with Create 2
Entered Execute 1
Done with Execute 1
Entered Execute 2
Done with Execute 2
Entered Finished 1
Waiting in Finished 1

Figure 69: Result after using “Skip until the current workflow stage”

While the user *perceives* the workflow skipping a stage, in reality it is going through the stages, but has the ability to logically skip the contents of that stage (provided you design it to do so).

The previous example seems to skip a stage, but the workflow actually goes through the stages. Figure 70 shows how you can design a feeder stage that enables the workflow to skip whole blocks of stages

Stage: Test Create Stage 1

Log [In Create 1](#) to the workflow history list

Transition to stage

If Project Web App starts the workflow normally or restarts the workflow and includes this stage

Go to [Test Execute Stage 1](#)

Else

Go to [Test Finish Stage 1](#)

Stage: Test Execute Stage 1

Log [In Execute 1](#) to the workflow history list

then Wait for [Event: When a project is submitted](#)

Transition to stage

Go to [Test Execute Stage 2](#)

Stage: Test Execute Stage 2

Log [In Execute 2](#) to the workflow history list

then Wait for [Event: When a project is submitted](#)

Transition to stage

Go to [Test Execute Stage 3](#)

Stage: Test Execute Stage 3

Log [In Execute 3](#) to the workflow history list

then Wait for [Event: When a project is submitted](#)

Transition to stage

Go to [Test Finish Stage 1](#)

Stage: Test Finish Stage 1

Log [In Finished 1](#) to the workflow history list

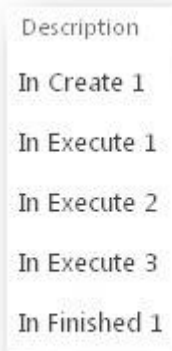
then Wait for [Event: When a project is submitted](#)

Transition to stage

Go to [End of Workflow](#)

Figure 70: Workflow example with skipping in transition

If we manually run the workflow, Figure 71 shows the results.



Description
In Create 1
In Execute 1
In Execute 2
In Execute 3
In Finished 1

Figure 71: Result of running with skipping in transition

If we skip to any stage other than Create 1, then we get the following:



Description
In Create 1
In Finished 1

Figure 72: Result of running to any stage other than Create 1

Unlike the previous example, we did not go through the other stages, but did a true skip. Unless more logic is designed into the workflow, the stage the user selects is not the one where the workflow finishes; rather, it is the stage defined in the `Else` statement (but the correct logic for skipping is depends on proper workflow design).

5.11 How to build your own custom actions to use in SharePoint Designer

For general information about building custom actions, see <http://msdn.microsoft.com/en-us/library/jj163911.aspx>

SharePoint Designer provides a collection of workflow actions that are available through the Workflow Designer user interface (UI). Although the range of workflow actions that are included in SharePoint Designer is extensive, it is nevertheless finite. In some cases, you may need to model a business process whose requirements are not met by the default library of workflow actions.

Because business processes often have specialized requirements, SharePoint 2013 can use custom workflow actions. You can develop custom actions by using Visual Studio, and then package and deploy them to SharePoint. At that point, the custom action becomes available to workflow authors in SharePoint Designer, just as if it were in the library of existing actions. This capability lets you

customize the functionality in your workflow authoring environment to match any of your specialized business processes.

5.11.1 Overview of a custom action

An action is a wrapper that abstracts the functionality of its underlying activity in SharePoint Designer. At run time, the underlying activity, not the action itself, is executed in the Windows Server AppFabric. In this sense, actions are just design-time abstractions of underlying functions in the SharePoint Designer workflow authoring environment (in addition to being elements of the SharePoint Designer UI).

Like all actions, custom actions are *web scoped*—that is, they are activated at the level of the SharePoint site collection, or **SharePoint.SPWeb** instance.

Actions are defined in XML definition files that have the `.actions4` file name extension. The underlying activity (or activities), on the other hand, are defined in a XAML file.

5.11.2 Writing custom activities in Visual Studio 2012

Visual Studio 2012 provides a **Workflow custom activity** item type within SharePoint projects. You can use the item type to create a custom activity that you can then import as a custom action in SharePoint Designer 2013.

5.11.3 Updating and deleting custom actions

After your custom action is deployed, you can update or remove it very easily. All you have to do is open the activity project in Visual Studio, make the changes that you want, and then package and redeploy as described in the preceding procedure. To remove the custom action, you can just uninstall the feature on the target site collection.

Feature activation Activating a custom action feature on a site collection (that is, on an **SPWeb** instance) succeeds only if the Windows Azure/ Workflow Manager Client 1.0 (the multitenant workflow engine) is correctly configured.

Two troubleshooting hints that may help ensure a correct configuration include:

- Go to the Site Features page and ensure that the feature that contains the custom action is activated.
- Query the Workflow Manager Client 1.0 database to ensure that the activity is successfully deployed.

For more information and see a full example to create, package, and deploy a custom activity see the MSDN article: How to: Build and deploy workflow custom actions: <http://msdn.microsoft.com/en-us/library/jj163911.aspx> .

There is also a custom action code sample available in the Samples library of MSDN:

<http://code.msdn.microsoft.com/SharePoint-2013-workflow-41e5c0f9>

6 USING SHAREPOINT LISTS TO CAPTURE DEMAND

A new scenario is now available to capture Demand Management information.

Instead of capturing demands using the PDP pages one by one, it is possible to first capture the demands in a SharePoint list that contains the fields necessary for the demand.

The SharePoint list can then be used directly to create the new project and follow the standard Demand Management workflow.

For more information and see the following blog article: Demand Management and Ideation in Project Online: <http://blogs.office.com/b/project/archive/2012/11/05/demand-management-and-ideation-in-microsoft-project-online.aspx>

6.1 Capture the demands in SharePoint List

You need first to create a custom SharePoint list with specific columns, which will be later mapped to the enterprise custom fields in your project.

The list of items to promote to projects can exist anywhere in the site collection where the Project Server service is enabled. For example if your PWA site is: <http://myserver/pwa>, and that is not also the Site Collection root, but rather the root is <http://myserver>, then the list could be in the parent, sibling or subweb of PWA (or even PWA itself).

The data is entered in the custom list using the standard SharePoint data entry mechanism.

6.2 Create an Enterprise Project from a SharePoint List

To create an enterprise project from a SharePoint list, select the items in the list that you want to promote:

In the **ITEM** tab of the ribbon, choose **Create Projects**. If you don't see the **Create Projects** button that means your list is not hosted in a supported site collection.

.

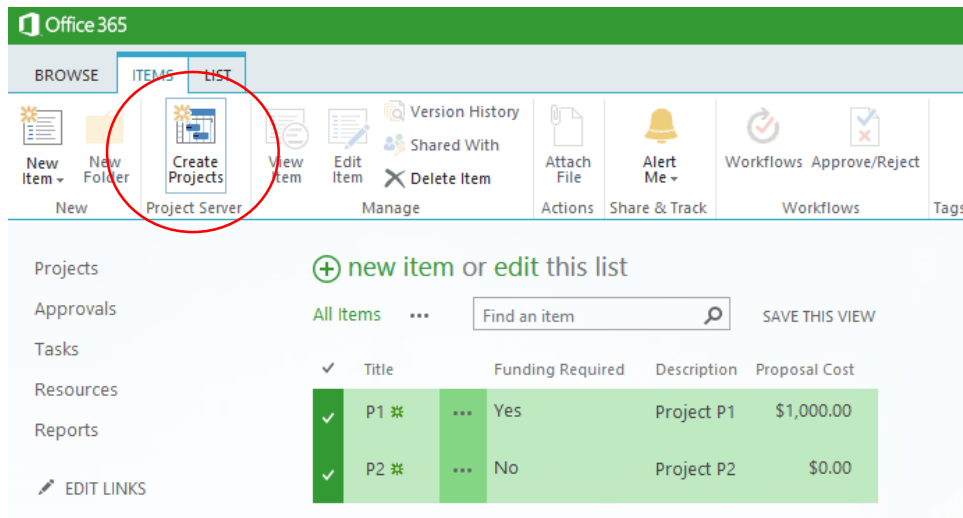


Figure 73: Create Projects from a SharePoint list

A dialog box opens where you can select the mapping to use between the SharePoint list columns and the Project Server custom fields, and select the enterprise project type.

Note: It is better to use a column of type Text Field or Multiple Line Text field, which enables them to be mapped to enterprise custom fields.

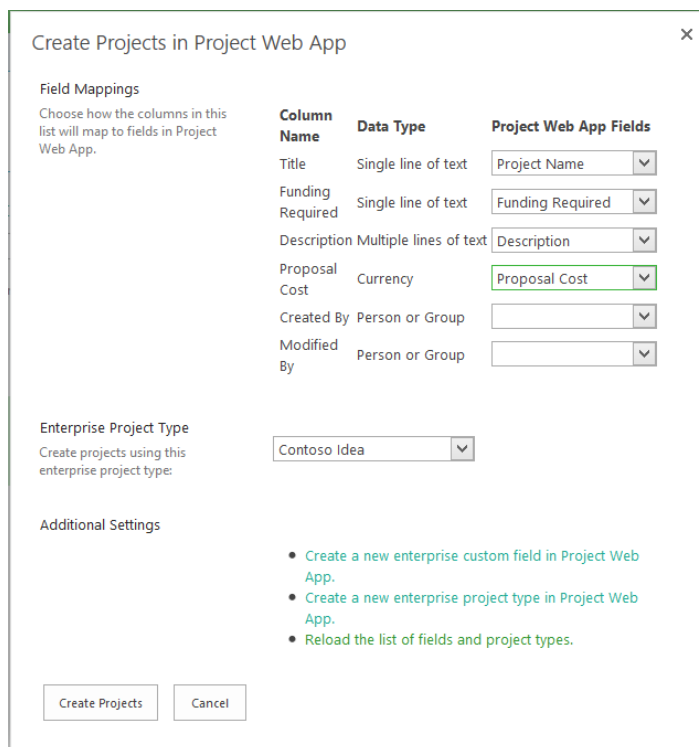


Figure 74: Create Projects dialog box to map list columns to Project Web App enterprise custom fields

Note that you can set the default mappings by navigating to the SharePoint list settings, and then choosing **Project Server settings** options (see Figure 75).

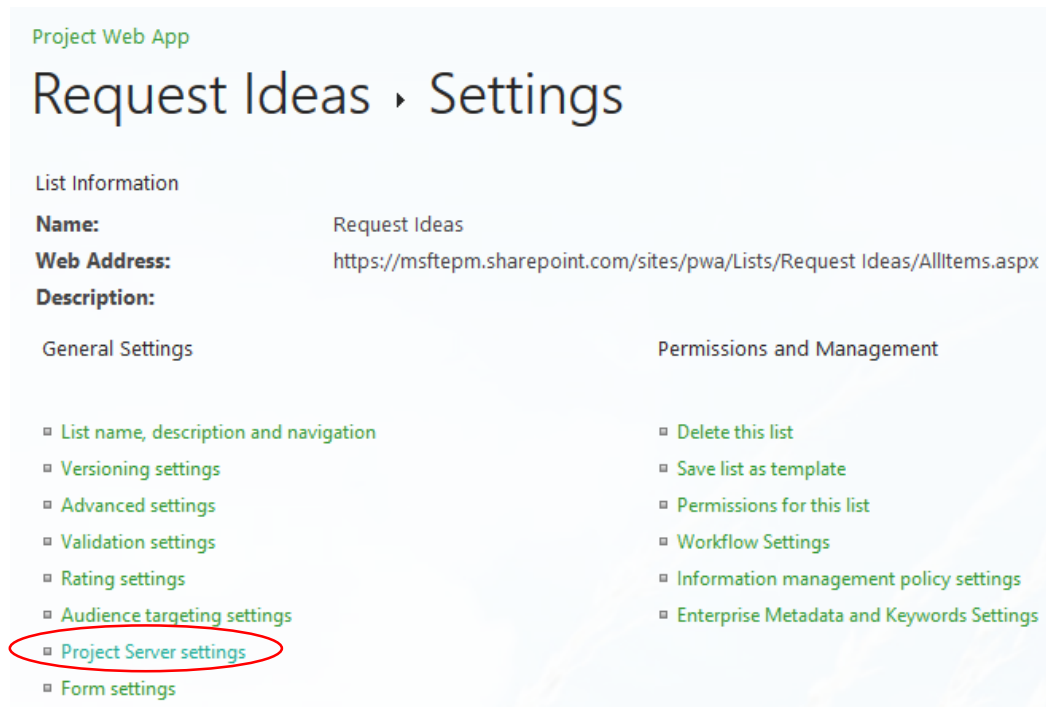


Figure 75: Project Server settings to define list mapping

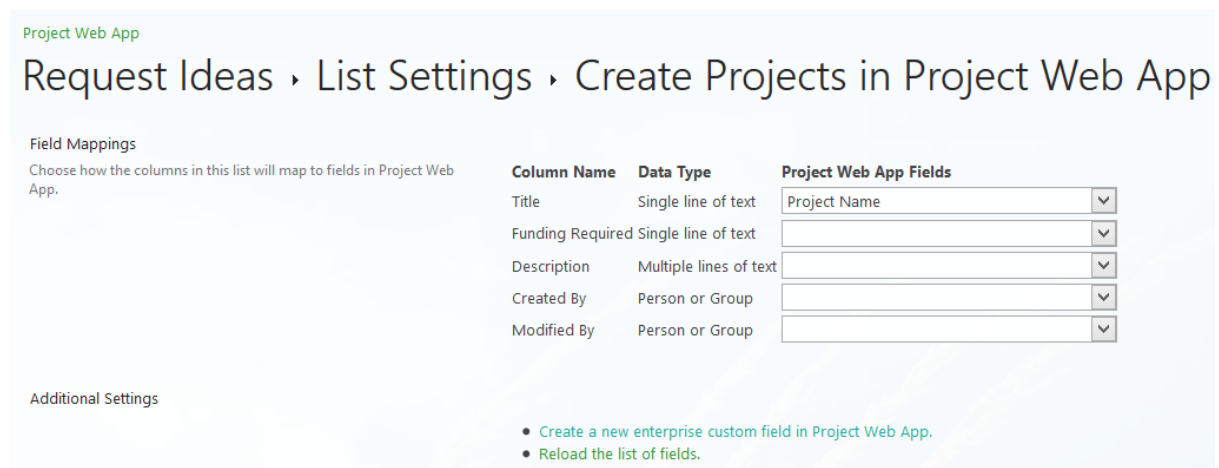


Figure 76: Field mappings column name to PWA Fields

When the projects are created, the dialog box confirms the successful creation (see Figure 77).

Create Projects in Project Web App

Field Mappings
Choose how the columns in this list will map to fields in Project Web App.

Column Name	Data Type	Project Web App Fields
Title	Single line of text	Project Name
Funding Required	Single line of text	Funding Required
Description	Multiple lines of text	Description
Created By	Person or Group	
Modified By	Person or Group	

Enterprise Project Type
Create projects using this enterprise project type: Contoso Idea

Additional Settings

- Create a new enterprise custom field in Project Web App.
- Create a new enterprise project type in Project Web App.
- Reload the list of fields and project types.

Congratulations (2 of 2) projects have been successfully created.

View Details: [View Details](#) [Close](#)

Figure 77: Dialog box after creation of projects

After a project is created, you can use the standard PWA Demand Management process to complete the workflows (see Figure 78).

Workflow Status

1-Create

s1-IdeaCollection

Current Workflow Stage: s1-IdeaCollection
Workflow Stage Status

Sections to be completed for this stage:
Pages below may require attention. Click "Next" in the ribbon above to scroll through each page or select an individual page by clicking on it below.

Page Name	Status	Description
Idea Collection	Complete	
Project Details	Complete	This section captures detail information about the project

Figure 78: Continue the standard workflow process

7 DEPLOY: DEPLOYMENT FROM DEV TO QA to PRODUCTION

After you have developed a workflow in your development environment, you can deploy it in a test environment and then a production environment.

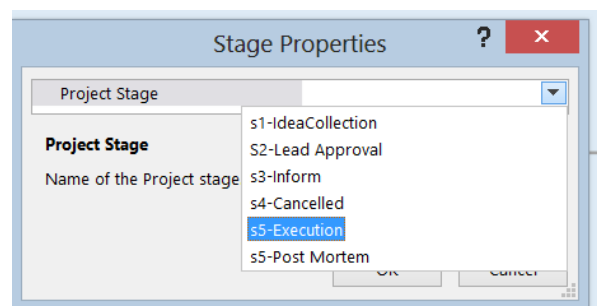
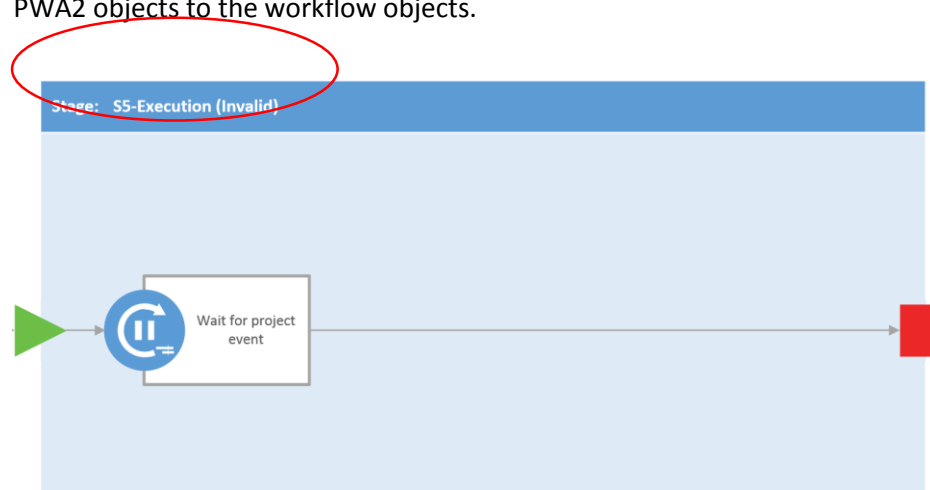
Workflows that are implemented by using SharePoint Designer are linked to the Project Web App instance that was opened with SharePoint Designer.

7.1 Copying a Workflow

You can export a workflow to Visio 2013, where the workflow was implemented with SharePoint Designer when it was connected to an instance named, for example, PWA1. Name this Visio diagram workflow: PWA1.vsd.

If you connect to another PWA instance (named PWA2) from SharePoint Designer and reimport the Visio file PWA1.vsd what happens?

SharePoint Designer tries to remap the different Project Server objects from PWA1 to PWA2. When it cannot find a valid mapping, it marks the stage as invalid. You can manually remap the correct PWA2 objects to the workflow objects.



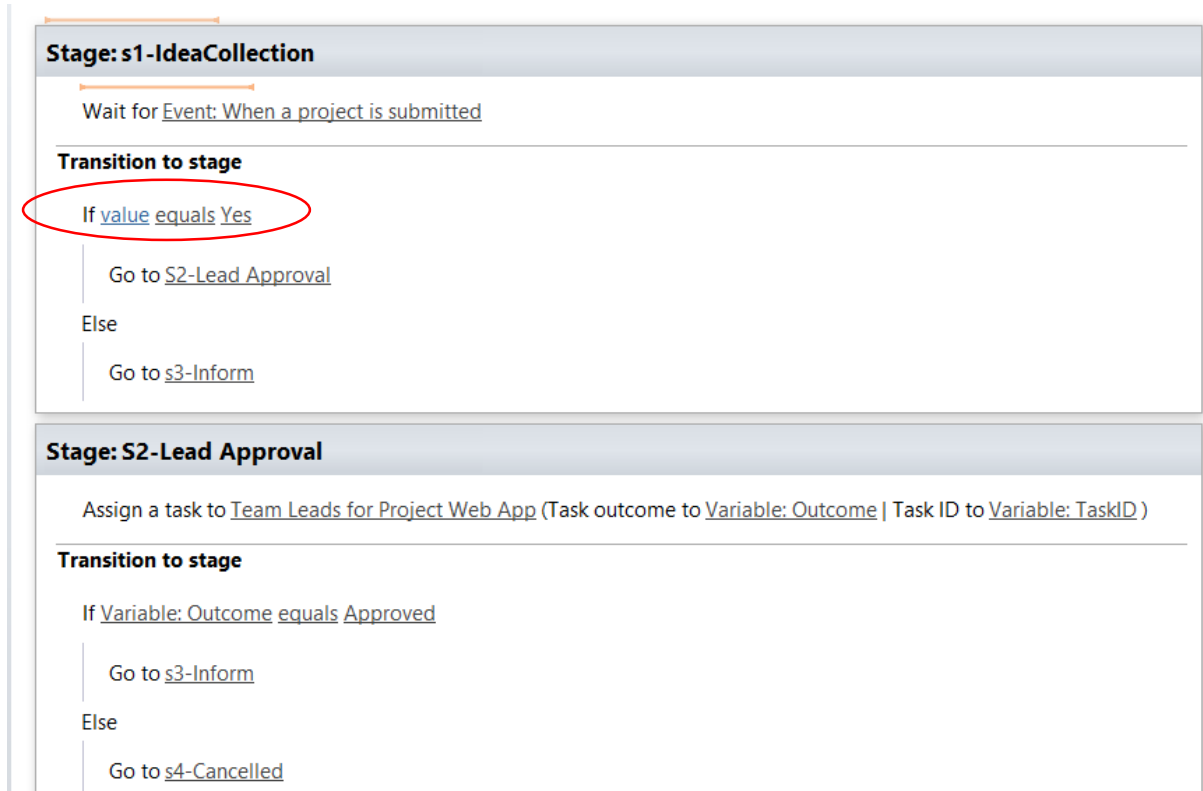


Figure 79: Remapping the PWA2 objects

If the objects have the same name, the remapping happens automatically. Otherwise the remapping must be done manually.

To copy the Project Server objects from the instance PWA1 to PWA2, you can either manually recreate them or you can use a 3rd party tool such as Fluent Pro:

<http://www.fluentpro.com/projectserver2013.html>

FluentPro FluentBooks 2013: <http://www.fluentpro.com/productsfluentbooks2013.html>

As of this writing, the Project Server 2010 Resource Kit tool: Playbook does not work with Project Server 2013.

8 DEBUG/MONITOR: TROUBLESHOOTING WORKFLOWS

This chapter shows how we can debug a workflow during development and also how to troubleshoot a workflow when it is running.

You cannot use the standard Visual Studio debugging tools to debug your workflow.

You can use the special Log History internal list to manually log information that can help your debugging. To log a message, use the **Log to History List** action (see Figure 80).

then Log [message](#) to the workflow history list

Figure 80: Logging a message to the workflow history list

The message can contain values of internal variables, as in Figure 81.

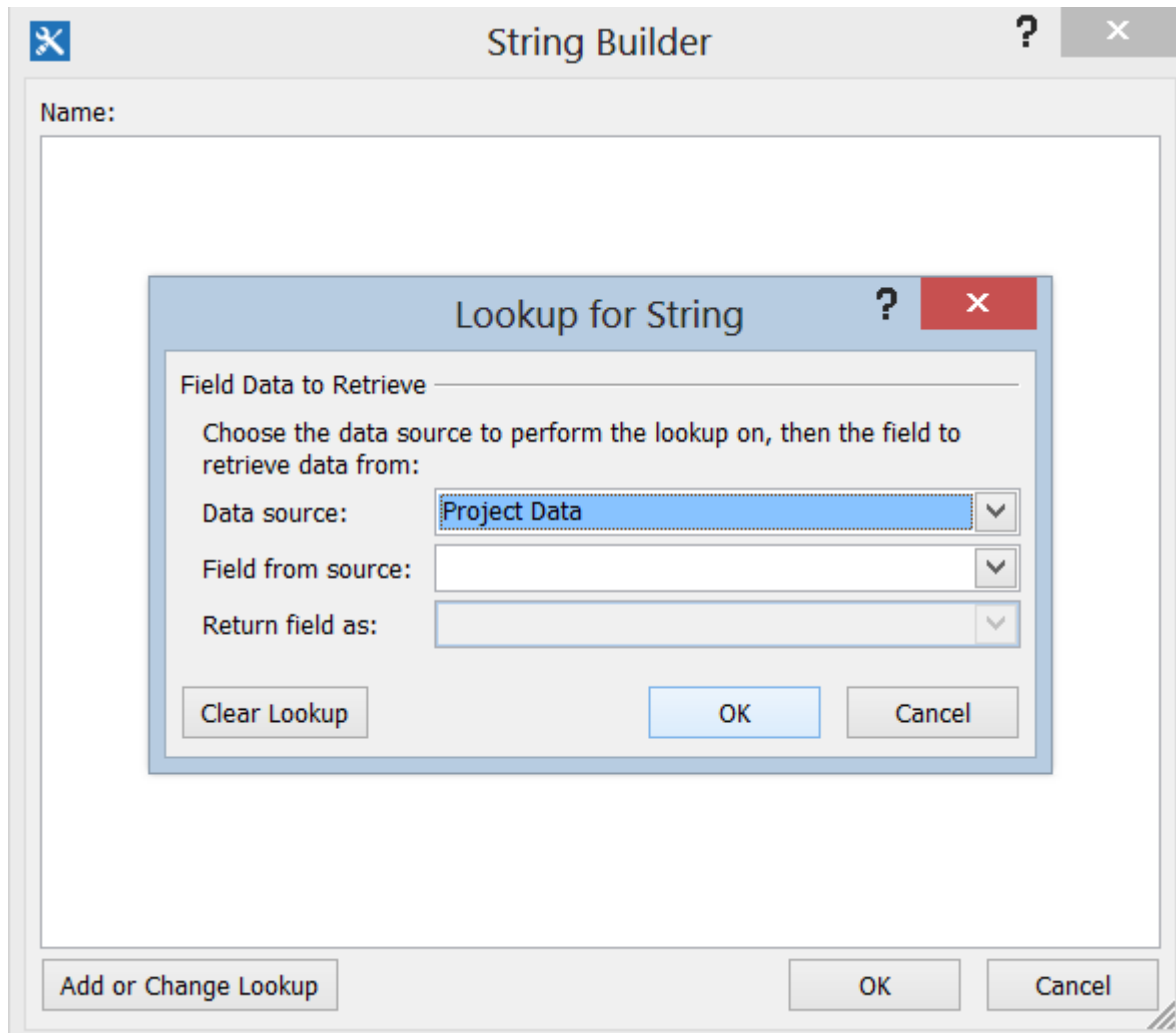


Figure 81: Formatting your message to display in Log History

8.1 Troubleshooting a Workflow

For information on how to troubleshoot a workflow, see the following article:

<http://blogs.technet.com/b/projectadministration/archive/2009/12/21/how-to-troubleshoot-yourworkflows.aspx>

- Check the workflow status page
- Check the ULS logs (Only for on-premises)

8.1.1 Check the workflow status page

Figure 82 shows two different ways to check the workflow status page, based on your requirements.

Actions	Screen																																																																		
<p>Check from within the project</p> <ul style="list-style-type: none">Open a project with a failing workflowGo to the Stage Status Page. (This is the first page in a workflow stage.)In the workflow status page, expand the All Workflow Stages section.	<div><p>Sections to be completed for this s</p><p>Pages below may require attention. Cli</p><table><tr><td></td><td>Page Name</td></tr><tr><td></td><td>Project Details</td></tr><tr><td></td><td>Schedule</td></tr></table><p>Pages marked with "*" require attention</p><div>All Workflow Stages</div></div> <p>This page allows you to check the status of your workflow and shows which stages were executed.</p>		Page Name		Project Details		Schedule																																																												
	Page Name																																																																		
	Project Details																																																																		
	Schedule																																																																		
	<div><p>All Workflow Stages</p><p>Workflow Stage status may not show current information until the project is either Submitted or the page is Refreshed.</p><table><tr><th></th><th>Stage</th><th>State</th><th>Entry Date</th><th>Completion Date</th><th>St</th></tr><tr><td></td><td>Workflow Phase: 1-Create</td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>s1-IdeaCollection</td><td>In Progress (Waiting for Input)</td><td>1/23/2013 8:07 AM</td><td>1/23/2013 8:07 AM</td><td></td></tr><tr><td></td><td>Workflow Phase: 2-Select</td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>s2-LeadApproval</td><td>Not Started</td><td></td><td></td><td></td></tr><tr><td></td><td>s3-Inform</td><td>Not Started</td><td></td><td></td><td></td></tr><tr><td></td><td>s4-Cancelled</td><td>Not Started</td><td></td><td></td><td></td></tr><tr><td></td><td>Workflow Phase: 3-Manage</td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>S5-Execution</td><td>Not Started</td><td></td><td></td><td></td></tr><tr><td></td><td>Workflow Phase: 4-Finished</td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>S6-PostMortem</td><td>Not Started</td><td></td><td></td><td></td></tr></table></div>		Stage	State	Entry Date	Completion Date	St		Workflow Phase: 1-Create						s1-IdeaCollection	In Progress (Waiting for Input)	1/23/2013 8:07 AM	1/23/2013 8:07 AM			Workflow Phase: 2-Select						s2-LeadApproval	Not Started					s3-Inform	Not Started					s4-Cancelled	Not Started					Workflow Phase: 3-Manage						S5-Execution	Not Started					Workflow Phase: 4-Finished						S6-PostMortem	Not Started			
	Stage	State	Entry Date	Completion Date	St																																																														
	Workflow Phase: 1-Create																																																																		
	s1-IdeaCollection	In Progress (Waiting for Input)	1/23/2013 8:07 AM	1/23/2013 8:07 AM																																																															
	Workflow Phase: 2-Select																																																																		
	s2-LeadApproval	Not Started																																																																	
	s3-Inform	Not Started																																																																	
	s4-Cancelled	Not Started																																																																	
	Workflow Phase: 3-Manage																																																																		
	S5-Execution	Not Started																																																																	
	Workflow Phase: 4-Finished																																																																		
	S6-PostMortem	Not Started																																																																	

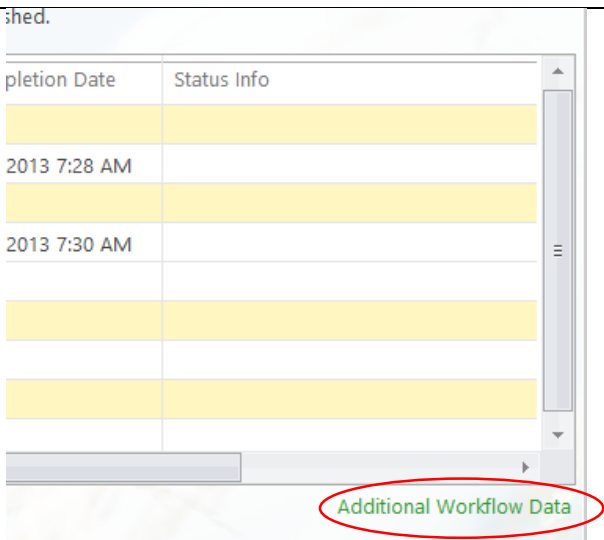
Actions	Screen
Click the Additional Workflow Data link which as the bottom right to get more information	 <p>The screenshot shows a table with two columns: 'Completion Date' and 'Status Info'. The table contains several rows, with the first two rows highlighted in yellow. The first row shows '2013 7:28 AM' and the second row shows '2013 7:30 AM'. At the bottom right of the table, there is a link labeled 'Additional Workflow Data' which is circled in red.</p>

Figure 82: Steps to troubleshoot a workflow

8.1.2 View the ULS logs (only available on-premises)

ULS logs are saved in the C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\15\LOGS directory.

Tips: Create a desktop shortcut to this location on the server. To help filter, sort, compare, and save parts of ULS logs, and to examine ULS logs from a remote computer, you can use a tool such as ULS Viewer. For information, see <http://archive.msdn.microsoft.com/ULSViewer>.

1. Logs are saved in separate files. Find the log with a time stamp as close to the time you are most concerned about, and open it.
2. For workflows, some key words to look for when going through the logs are:
 - SharePoint Foundation
 - Startworkflow
 - Winwf
 - entering...activity
 - leaving... activity
3. You can increase the logging level on the following categories from the central administration: http://servername:Port/_admin/metrics.aspx
 - Project Server: Project Server Workflow
 - SharePoint Foundation: Workflow Infrastructure

9 REFERENCES

9.1 General references

Title	URL Reference
Project Web site	http://www.microsoft.com/project
Project Server for IT pros TechCenter (TechNet)	http://technet.microsoft.com/projectserver
Project developer center (MSDN)	http://msdn.microsoft.com/Project
Project 2013 SDK download	https://www.microsoft.com/en-us/download/details.aspx?id=30435
Microsoft Project Conference 2012 Video content	www.microsoft.com/showcase/en/US/channels/microsoftproject

Blogs	URL Reference
Official Blog of the Product Development	http://blogs.office.com/b/project/
Project Developers	http://blogs.msdn.com/b/project_programmability
Project IT Pro	http://blogs.technet.com/b/projectadministration
Project Server 2013 SDK	https://www.microsoft.com/en-us/download/details.aspx?id=30435

Additional questions? See the Project forums!

<http://social.msdn.microsoft.com/Forums/en-US/category/projectserver2010,projectprofessional2010>

9.2 References in this document

Document/Blogs	URL
Workflow development with SharePoint Designer or Visio 2013	http://msdn.microsoft.com/en-us/library/jj163272?v=(office.15).aspx
Getting started developing Project Server 2013 workflows	http://msdn.microsoft.com/en-us/library/ee767694?v=office15.aspx
Creating Project Workflows using Visual Studio 2012	http://blogs.msdn.com/b/project_programmability/archive/2012/11/07/creating-project-workflows-using-visual-studio-2012.aspx
Workflow development with SharePoint Designer or Visio 2013	http://msdn.microsoft.com/en-us/library/jj163272?v=(office.15).aspx
Develop SharePoint 2013 workflows using Visual Studio	http://msdn.microsoft.com/en-us/library/jj163199(v=office.15).aspx
How to: Build and deploy workflow custom actions	http://msdn.microsoft.com/en-us/library/jj163911.aspx
Demand Management and Ideation in Project Online	http://blogs.office.com/b/project/archive/2012/11/05/demand-management-and-ideation-in-microsoft-project-online.aspx
Getting started developing Project Server 2013 workflows	http://msdn.microsoft.com/en-us/library/ee767694(office.15).aspx