Sitecore CMS 6

Workflow Cookbook

Tips and Techniques for CMS Administrators who design workflows

Table of Contents

Chapter 1

Introduction

This cookbook provides techniques for CMS Editors and Administrators who create and configure workflows.

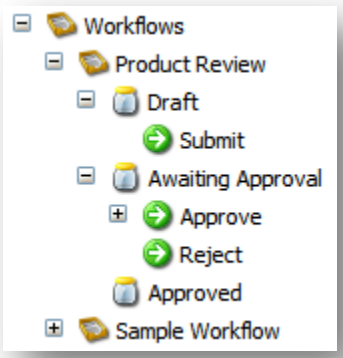
This document contains the following chapters:

- Chapter 1 — Introduction
- Chapter 2 — How to create a workflow?
- Chapter 3 — Restricting Access to Workflow States and Commands
- Chapter 4 — Using Workflow Actions

Chapter 2

How to create a workflow?

In this chapter we will create the workflow which has the following states, actions and commands:



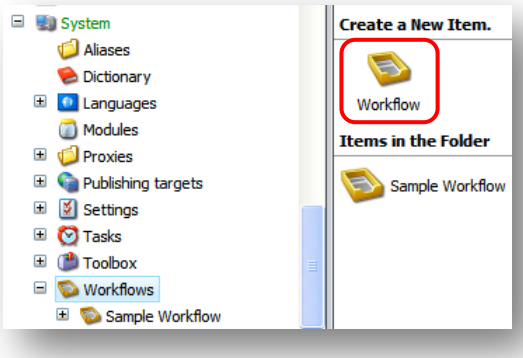
The procedure of creating a workflow involves the following steps:

- Creating a workflow definition item.
- Creating workflow states.
- Adding workflow actions or commands to the workflow states.
- Specifying the final state.

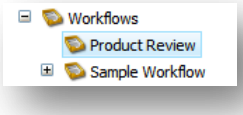
When the workflow is created you should assign it to a template to start using it.

2.1 Creating a Workflow Definition Item

To create a workflow definition item, navigate to **/sitecore/system/Workflows** and create a new workflow definition item using the Workflow template.



Call the new workflow “Product Review”:

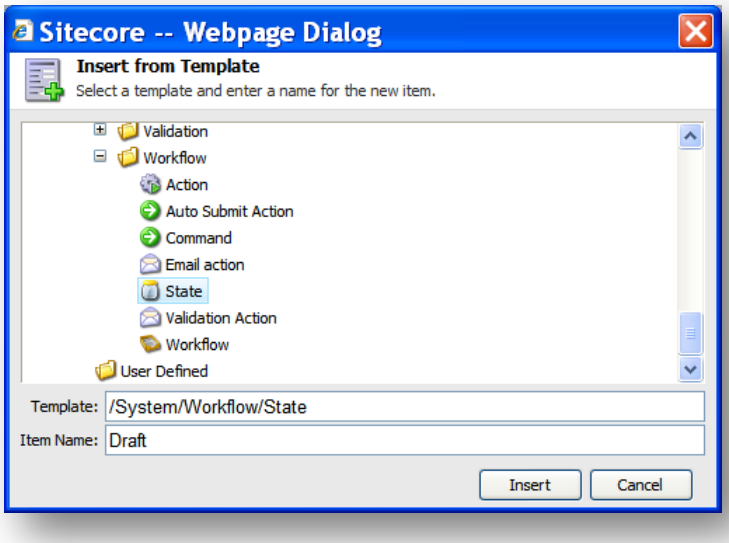


Leave the **Initial State** field blank yet.

2.2 Creating a Workflow State

To create a workflow state:

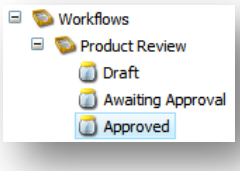
1. Select the workflow definition item.
2. Select the **Insert from Template** command and create the new workflow state using the **/System/Workflow/State** template. Call the first state “Draft”:



Click Insert.

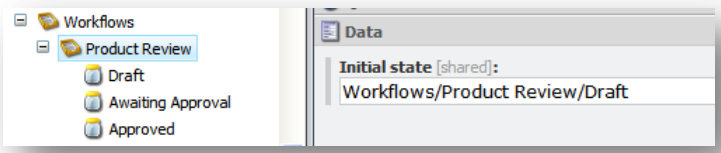
3. The state is created:

Create two more workflow states – “Awaiting Approval” and “Approved”. You should have the following:



2.2.1 Setting the Initial State

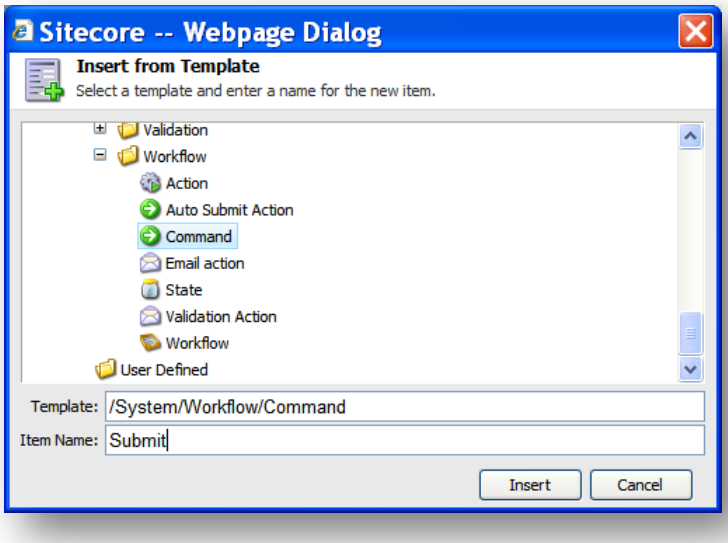
Now when you have the workflow states, select the workflow definitions item (Product Review) and set the **Initial State** field to Draft:



2.3 Adding Workflow Commands to Workflow States.

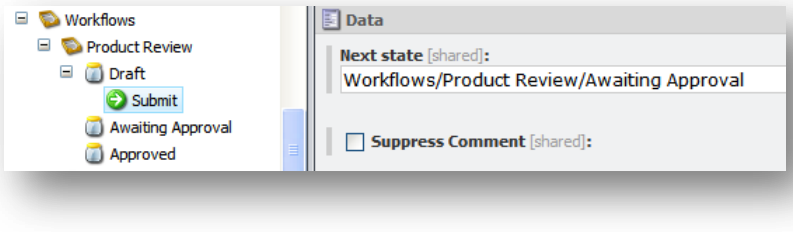
To add a command to a workflow state:

1. Select a workflow state (Draft in our example).
2. Select the **Insert from Template** command and create the new workflow command using the **/System/Workflow/Command** template. Call the new command “Submit”:



Click Insert.

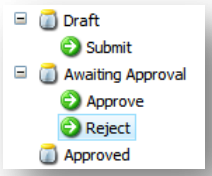
3. The command is created. In the “Next state” field of the command select the *Awaiting Approval* state:



Also create the following commands under Awaiting Approval state:

Approve with the Next State set to *Approved*.

Reject, with the Next State set to *Draft*.

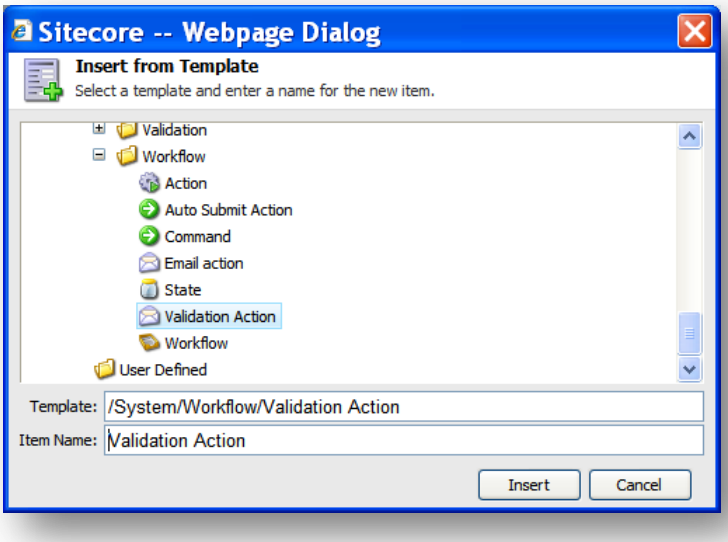


2.4 Adding Workflow Actions to States or Commands

In this example we will add a validation action to a workflow command.

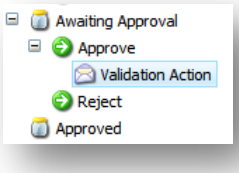
To add a workflow validation action to a workflow command:

1. Select a workflow command (Approve in our example).
2. Select the **Insert from Template** command and create the new workflow action using the **/System/Workflow/Validation Action** template. Call the new command "Validation Action":



Click Insert.

3. The validation action is added:



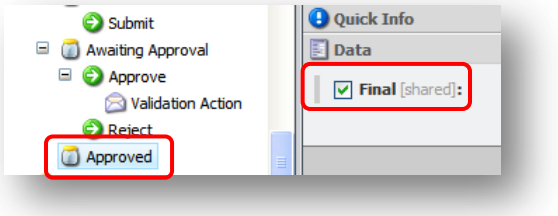
Now we should fill its fields.

4. In the Type field, enter the following:
`Sitecore.Workflows.Simple.ValidatorsAction,Sitecore.Kernel.`
5. In the Max Result Allowed field enter `Warning`.
6. Fill the Error result fields with the messages you want a user to see in case of validation errors, for instance, "You cannot approve an item with validation errors". For more information about validation settings, see the Client Configuration Cookbook, Chapter 3, Data Validation.

In a similar way, add the *Auto Publish* action to the *Workflow* state.

2.5 Defining the Final State

Select the state which is supposed to be the final state in your workflow and select the **Final** checkbox in the Data field:



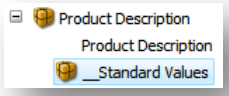
You will be able to publish only those items which are in the Approved state in this example (this does not apply to items which are not in a workflow of course).

2.6 Assigning a Workflow to a Template

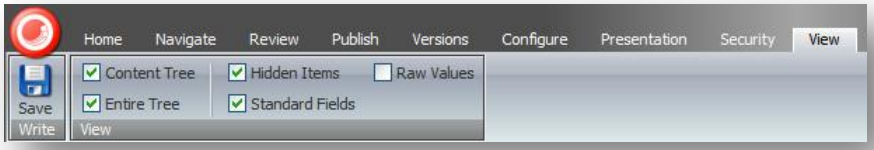
Once the workflow is created you should assign it to a template so that this workflow can be used.

The best practice is to assign a workflow on the Standard Values item of a template:

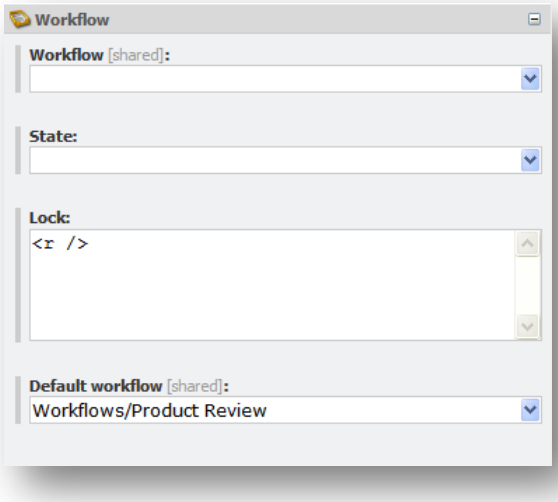
1. Select a template.
2. Select the *__Standard Values* item.



3. In the View tab, in the View group, select the Standard Fields checkbox.



4. In the **Workflow** section, in the **Default workflow** field select your workflow definition item.



   Leave the Workflow and the State fields blank, they will be filled automatically upon item creation basing on the Workflow settings. Leave the Lock field as it is.

5. Save the template and it is ready to use with workflow support.

Chapter 3

Restricting Access to Workflow States and Commands

This chapter describes how to restrict access to workflow states and commands.

3.1 Hiding a Workflow Command for Certain Users

The Content Editor and Workbox only displays workflow commands for non-Administrator users when:

- The user has write access to the associated item.

  and

- The user has write access to the command's parent workflow state.

  and

- The user has read access to the workflow command itself.

If you configure the Sitecore security settings so that a user does not meet one of these criteria, you will hide the workflow command from that user.

If the user must have write access to both the item and the workflow state, there are two ways to deny them read access to the workflow command.

- Turn off the inheritance access right for the workflow command item and do not grant read access to the workflow command to the user and all the roles that the user is a member of.
- Deny read access for the workflow command item to the user or one of the roles that the user is a member of.

Each of these approaches has its advantages and disadvantages.

- Turning off the inheritance access right means that you must explicitly grant access to all the roles that should be able to see the workflow state in the Workbox. This is the best approach when only a small number of users and roles need to see the workflow state in the Workbox.
- In the Sitecore security system, deny always overrules allow. When you explicitly deny a role read access, you can inadvertently prevent a user who has been assigned many roles from seeing the workflow item. Denying read access can have unanticipated results.

In general, we recommend that you turn off the inheritance access right and explicitly allowing access rights when the number of roles that require access is manageable.

3.2 Hiding a Workflow State in for Certain Users

Users who have read access to a workflow state can see that state in their workbox as long as the state includes workflow commands for which they have command execute access rights. If business requirements state that a particular workflow state should be hidden from a given set of users, you can restrict access to that state for those users by:

- Hiding all the workflow commands in the state from the users in question.

  or

- Explicitly hiding the workflow state itself from the users in question.

To explicitly hide a workflow state:

- Turn off the inheritance access right for the workflow state item and do not grant read access to the workflow state to the user and all the roles assigned to the user.

  or

- Deny the user or one of the roles that the user is assigned read access to the workflow state item.

Each of these approaches has its advantages and disadvantages.

- Turning off the inheritance access right means that you must explicitly grant access to all the roles that should be able to see the workflow state in the Workbox. This is the best approach when only a small number of users and roles need to see the workflow state in the Workbox.
- In the Sitecore security system, deny always overrules allow. When you explicitly deny a role read access, you can inadvertently prevent a user who has been assigned many roles from seeing the workflow item. Denying read access can have unanticipated results.

In general, we recommend that you turn off the inheritance access right and explicitly allowing access rights when the number of roles that require access is manageable.