Workflow

In almost all the Sitecore projects, developers have to create, sometimes customize and need to apply workflow to items. This guide covers all the possible scenarios and customization technic with code used for workflow. It includes what is workflow, how to create custom workflow, customizing comment window, customizing workbox and customizing items in workbox.

What is workflow?

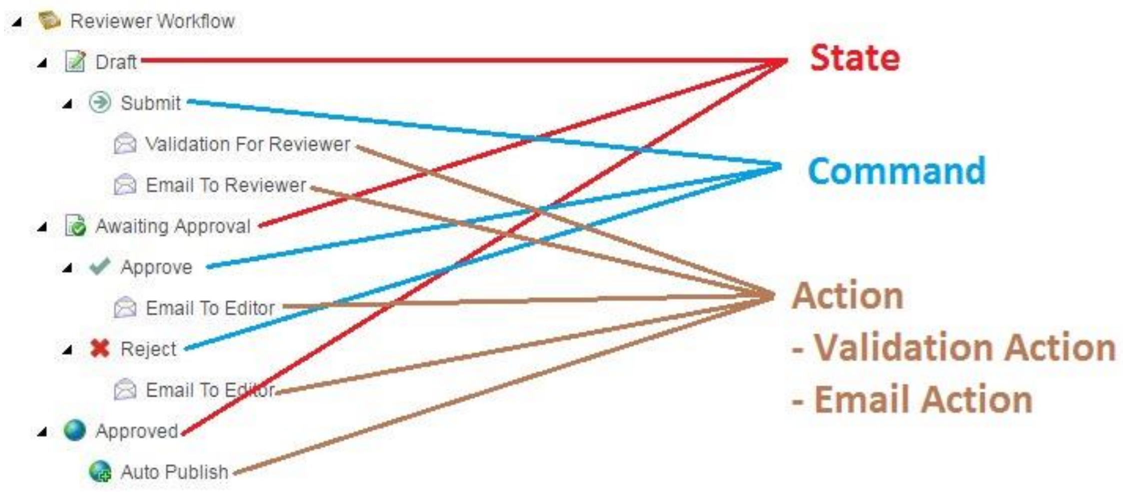
According to Sitecore "*Workflows ensure that items move through a predefined set of states before they become publishable, usually intended to ensure that content receives the appropriate reviews and approvals before publication to the live Web site.*"

All workflow are resides in **"/sitecore/system/Workflows/**" folder.

A workflow have states, commands and actions.

Every workflow start with draft state and end with publish or final state.

Most of the workflow looks indicated in below diagram



Let's deep dive into understanding Sitecore workflow with one real life requirement and its implementation.

Requirement:

Client will need the following roles:

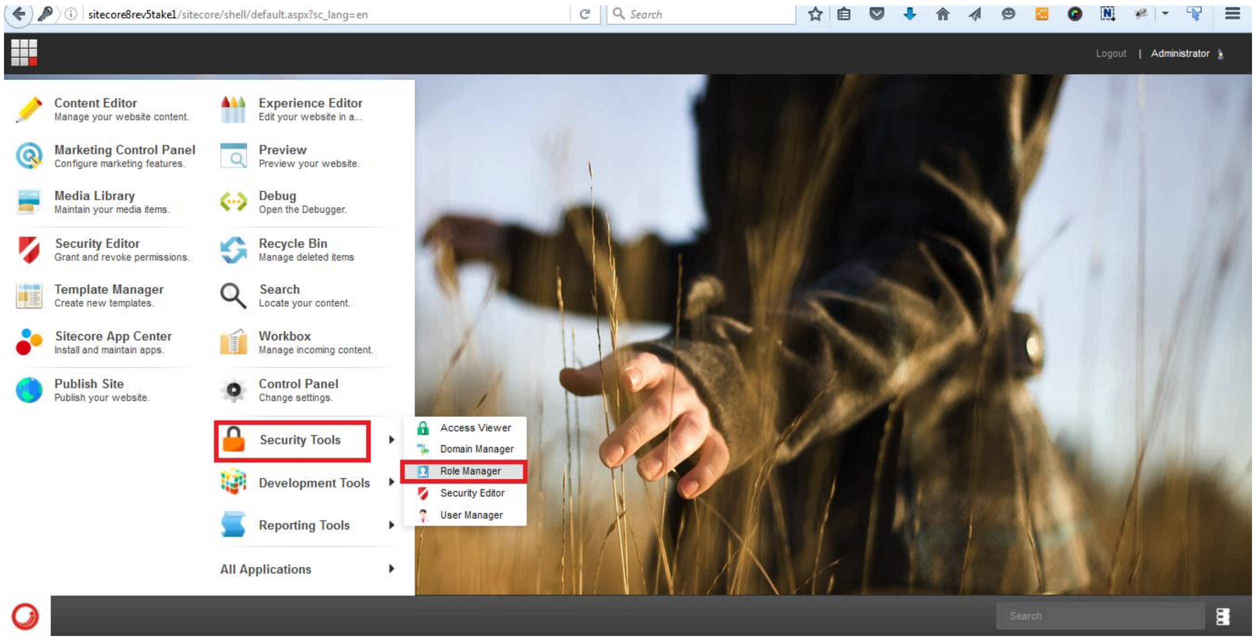
Editor – Can edit content but cannot publish content.

Reviewer – Can edit and publish content

Create Editor and Reviewer Roles

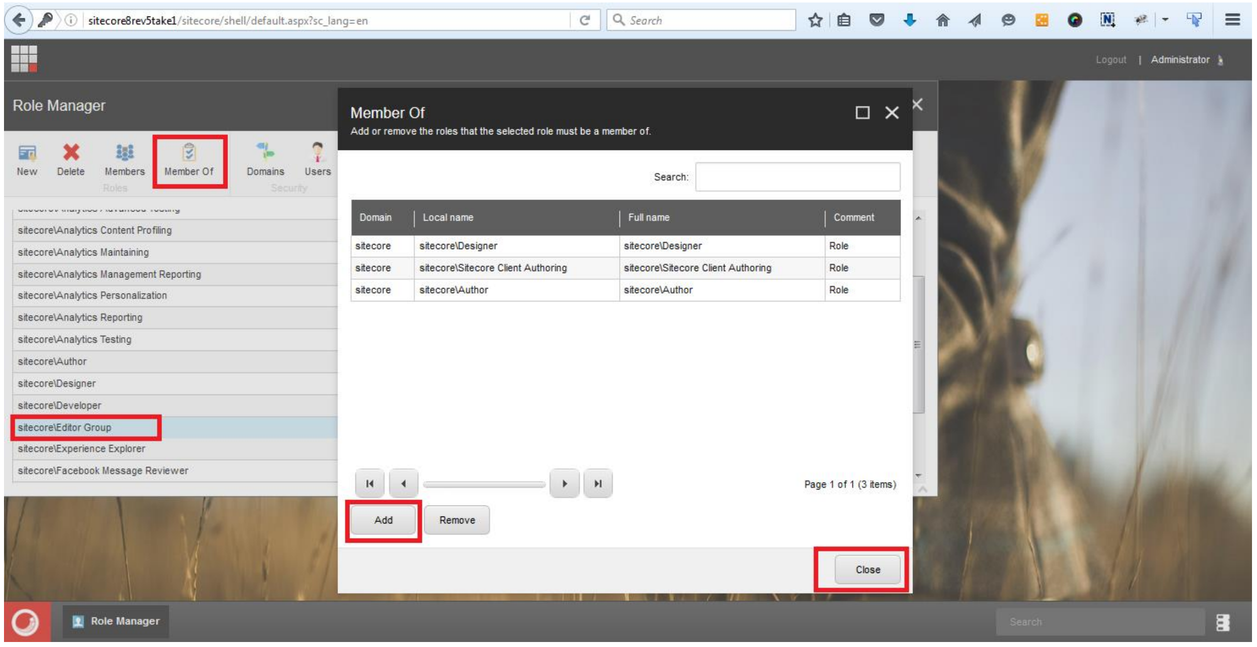
We will create two role in Sitecore namely – **Editor Group** and **Reviewer Group**

Logged in into Sitecore desktop and select **Security Tools** -> **Role Manager**



Create two new role - **Editor Group**, **Reviewer Group**

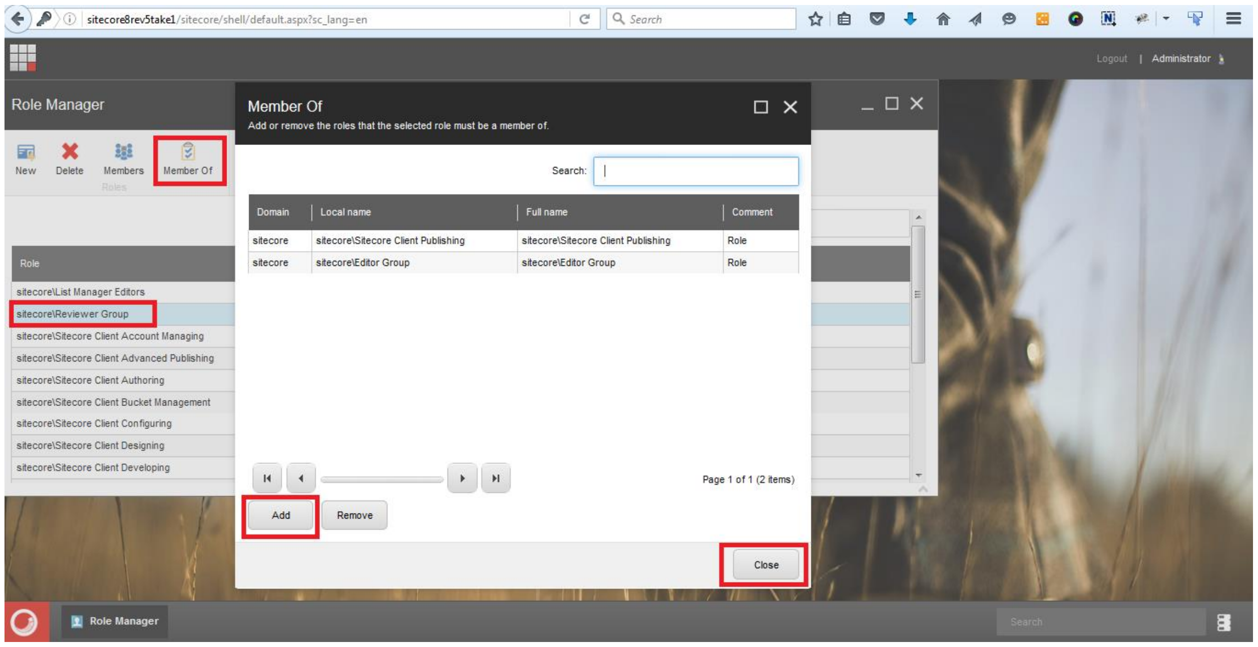
Assign editor role to **Editor Group** as


sitecore8rev5take1/sitecore/shell/default.aspx?sc_lang=en
Logout | Administrator
Role Manager
New
Delete
Members
Member Of
Roles
Domains
Users
Security
sitecore\Analytics Content Profiling
sitecore\Analytics Maintaining
sitecore\Analytics Management Reporting
sitecore\Analytics Personalization
sitecore\Analytics Reporting
sitecore\Analytics Testing
sitecore\Author
sitecore\Designer
sitecore\Developer
sitecore\Editor Group
sitecore\Experience Explorer
sitecore\Facebook Message Reviewer
Member Of
Add or remove the roles that the selected role must be a member of.
Search:
Page 1 of 1 (3 items)
Add
Remove
Close
Role Manager
Search

Domain	Local name	Full name	Comment
sitecore	sitecore\Designer	sitecore\Designer	Role
sitecore	sitecore\Sitecore Client Authoring	sitecore\Sitecore Client Authoring	Role
sitecore	sitecore\Author	sitecore\Author	Role

Assign reviewer role to **Reviewer Group**. Technically Reviewer is one who have editor as well as publishing permissions. So instead of assigning individual roles to Reviewer group you can directly assign roles as

Reviewer = Editor Group Role + Sitecore Client Publishing



Member Of

Add or remove the roles that the selected role must be a member of.

Search:

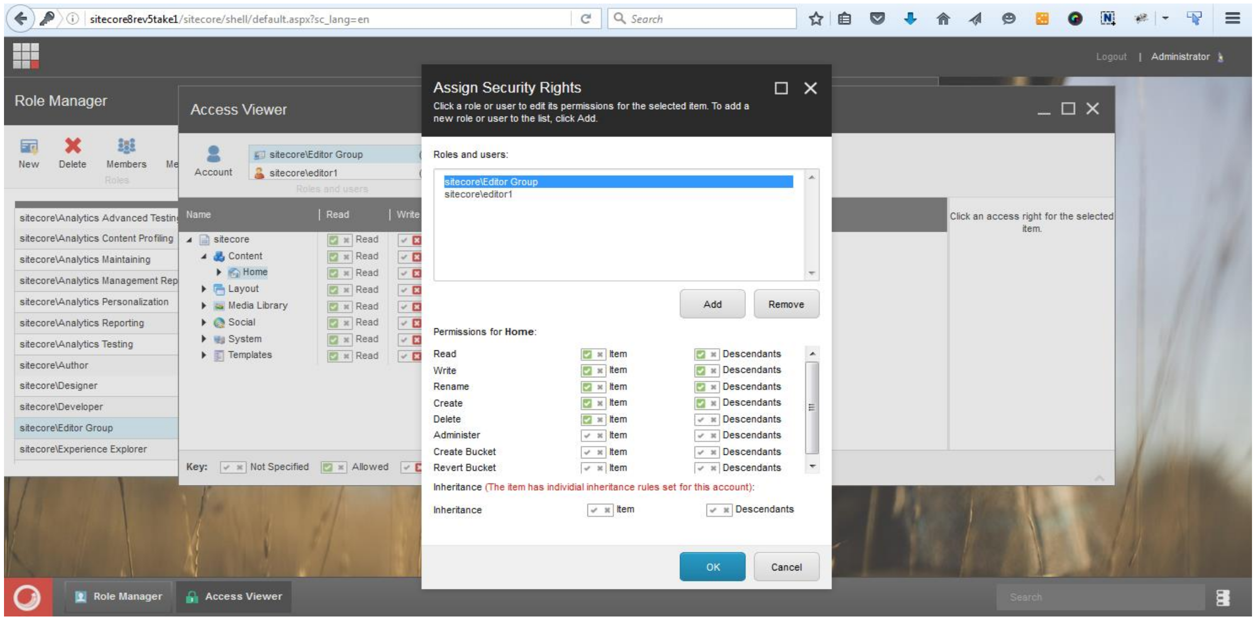
Domain	Local name	Full name	Comment
sitecore	sitecore\Sitecore Client Publishing	sitecore\Sitecore Client Publishing	Role
sitecore	sitecore\Editor Group	sitecore\Editor Group	Role

Page 1 of 1 (2 items)

Role
sitecore\List Manager Editors
sitecore\Reviewer Group
sitecore\Sitecore Client Account Managing
sitecore\Sitecore Client Advanced Publishing
sitecore\Sitecore Client Authoring
sitecore\Sitecore Client Bucket Management
sitecore\Sitecore Client Configuring
sitecore\Sitecore Client Designing
sitecore\Sitecore Client Developing

1. Assign permissions on items to roles

Assign permissions to these newly created role on content item.
First assign permissions to **Editor Group** Role. Note that we are NOT providing Delete Descendants permission to this role.



Assign Security Rights

Click a role or user to edit its permissions for the selected item. To add a new role or user to the list, click Add.

Roles and users:

sitecore\Editor Group
sitecore\editor1

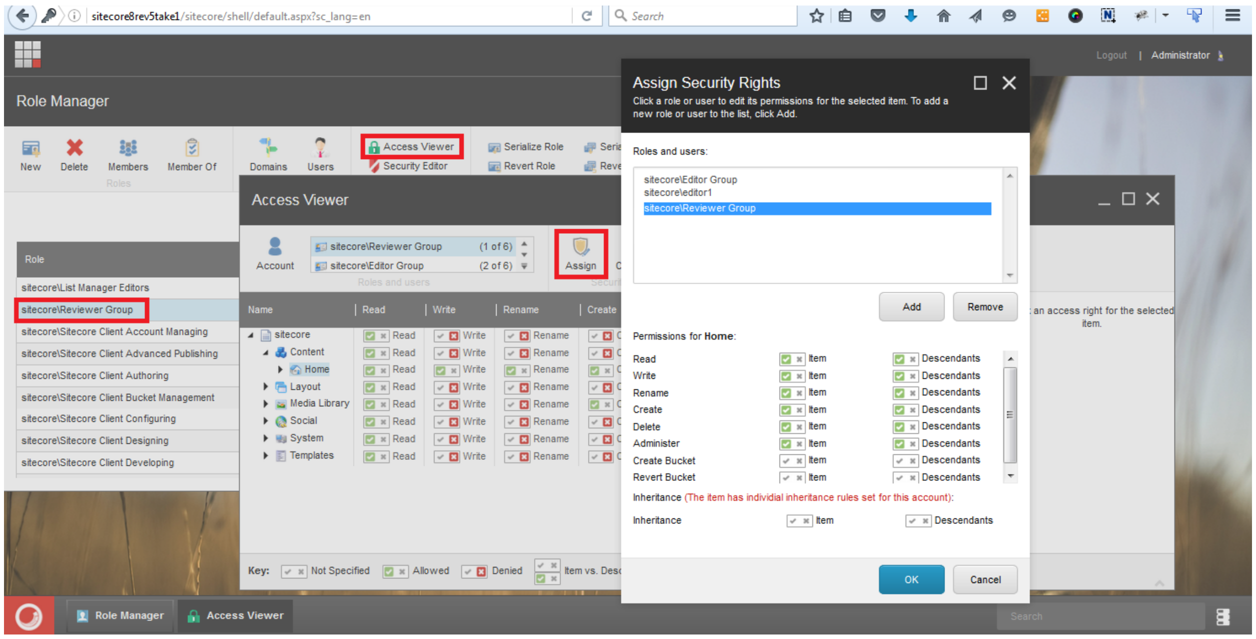
Permissions for Home:

Read	✓ Item	✓ Descendants
Write	✓ Item	✓ Descendants
Rename	✓ Item	✓ Descendants
Create	✓ Item	✓ Descendants
Delete	✓ Item	Descendants
Administer	Item	Descendants
Create Bucket	Item	Descendants
Revert Bucket	Item	Descendants

Inheritance (The item has individial inheritance rules set for this account):

Inheritance Item Descendants

Similarly assign permissions to **Reviewer Group** Role. Note that we are providing Delete Descendants as well as Administer level permission to this role.



Assign Security Rights

Click a role or user to edit its permissions for the selected item. To add a new role or user to the list, click Add.

Roles and users:

sitecore\Editor Group
sitecore\editor1
sitecore\Reviewer Group

Permissions for Home:

Read	✓ Item	✓ Descendants
Write	✓ Item	✓ Descendants
Rename	✓ Item	✓ Descendants
Create	✓ Item	✓ Descendants
Delete	✓ Item	✓ Descendants
Administer	✓ Item	✓ Descendants
Create Bucket	Item	Descendants
Revert Bucket	Item	Descendants

Inheritance (The item has individial inheritance rules set for this account):

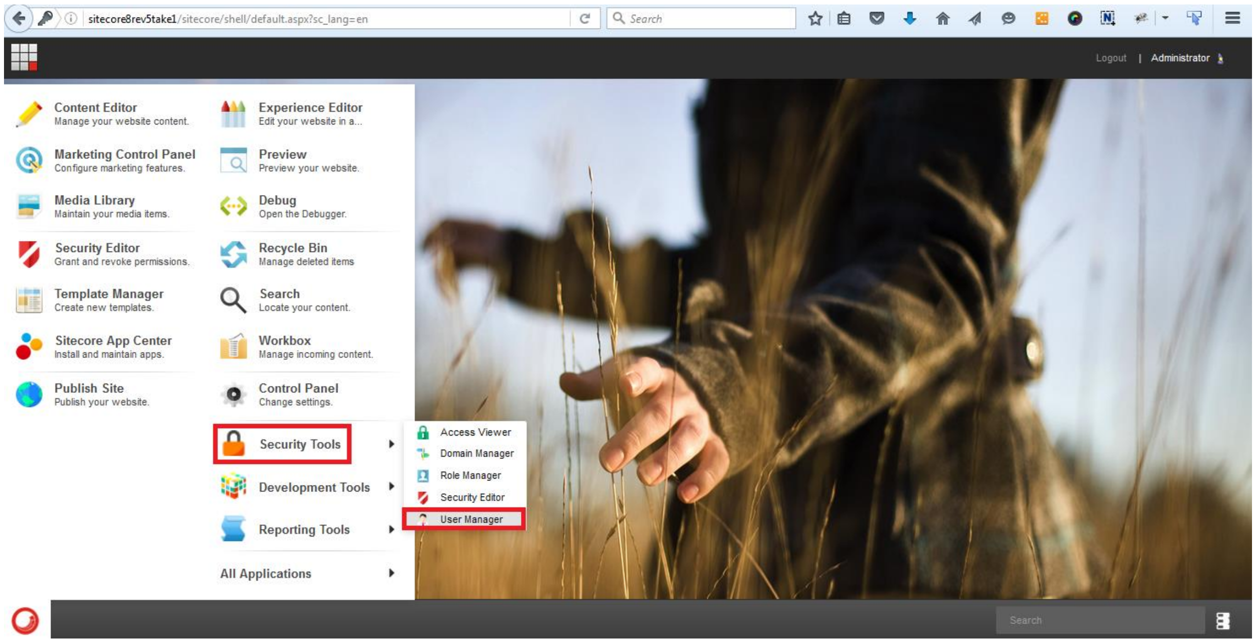
Inheritance Item Descendants

Role Manager
Access Viewer
Security Editor
Assign
sitecore\Reviewer Group (1 of 6)
sitecore\Editor Group (2 of 6)
OK
Cancel

Tips: - If you are not assigning item access permission to Roles/Users, then that user will not able to see workflow pending items in workbox.

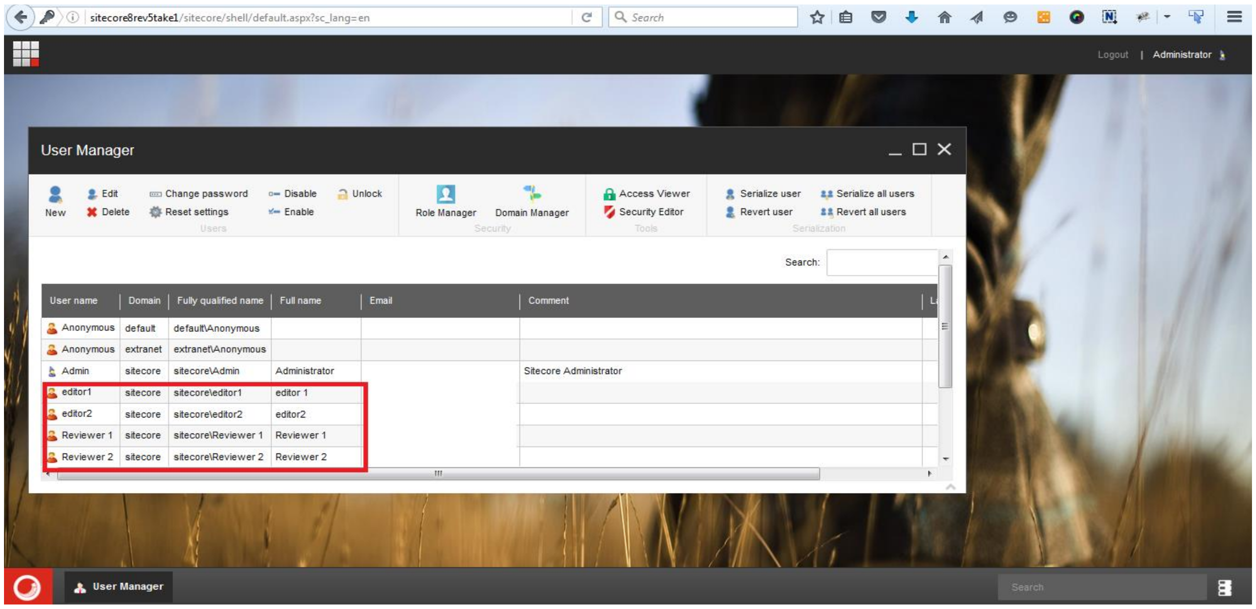
2. Create editor and reviewer users

Once we have created roles, now it’s time to create users under these roles. Logged in into Sitecore desktop and select **Security Tools** -> **User Manager**



Create some new users who play part of editors and reviewers.
For example, we created 4 users like

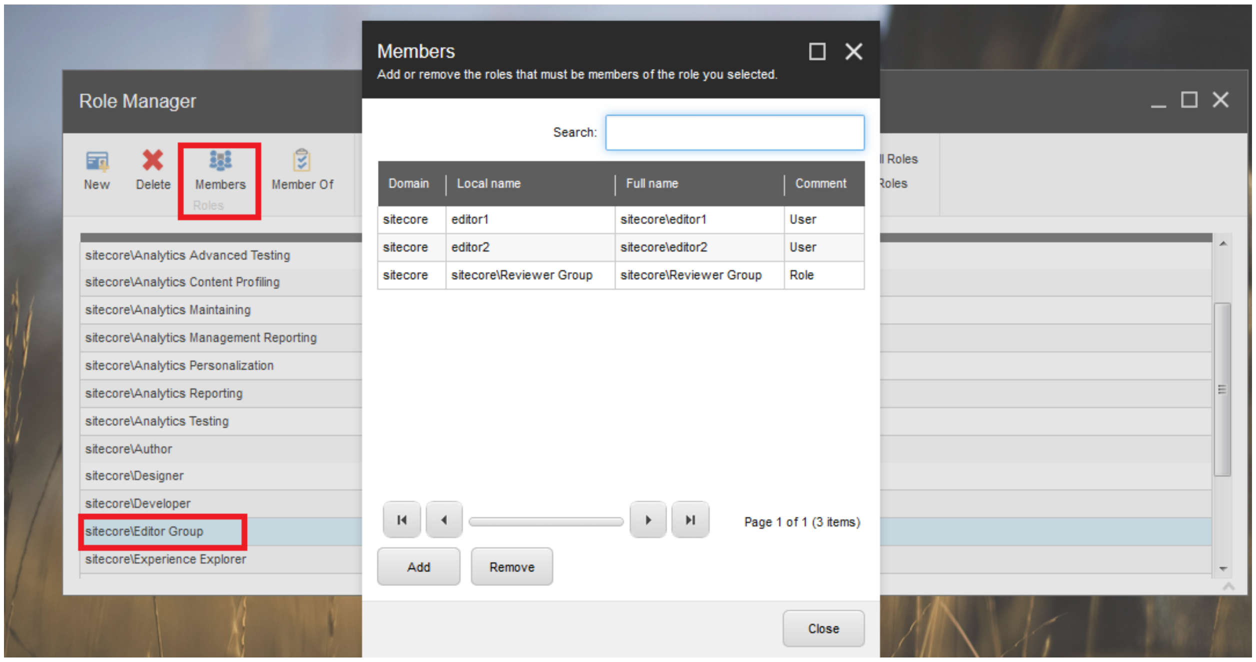
- Editor1
- Editor2
- Reviewer1
- Reviewer2



User name	Domain	Fully qualified name	Full name	Email	Comment
Anonymous	default	default\Anonymous			
Anonymous	extranet	extranet\Anonymous			
Admin	sitecore	sitecore\Admin	Administrator		Sitecore Administrator
editor1	sitecore	sitecore\editor1	editor 1		
editor2	sitecore	sitecore\editor2	editor2		
Reviewer 1	sitecore	sitecore\Reviewer 1	Reviewer 1		
Reviewer 2	sitecore	sitecore\Reviewer 2	Reviewer 2		

3. Map users with roles

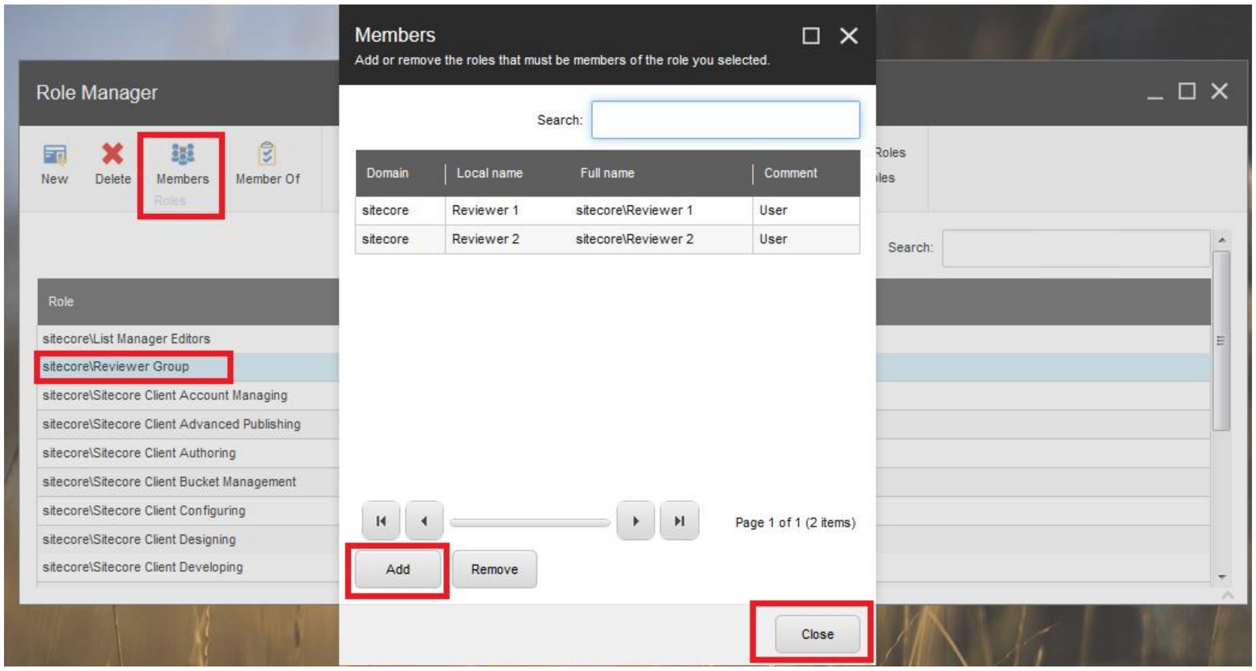
Assign user Editor1 and Editor2 to **Editor Group**.



Domain	Local name	Full name	Comment
sitecore	editor1	sitecore\editor1	User
sitecore	editor2	sitecore\editor2	User
sitecore	sitecore\Reviewer Group	sitecore\Reviewer Group	Role

sitecore\Analytics Advanced Testing
sitecore\Analytics Content Profiling
sitecore\Analytics Maintaining
sitecore\Analytics Management Reporting
sitecore\Analytics Personalization
sitecore\Analytics Reporting
sitecore\Analytics Testing
sitecore\Author
sitecore\Designer
sitecore\Developer
sitecore\Editor Group
sitecore\Experience Explorer

Similarly assign user Reviewer 1 and Reviewer 2 in **Reviewer Group**.

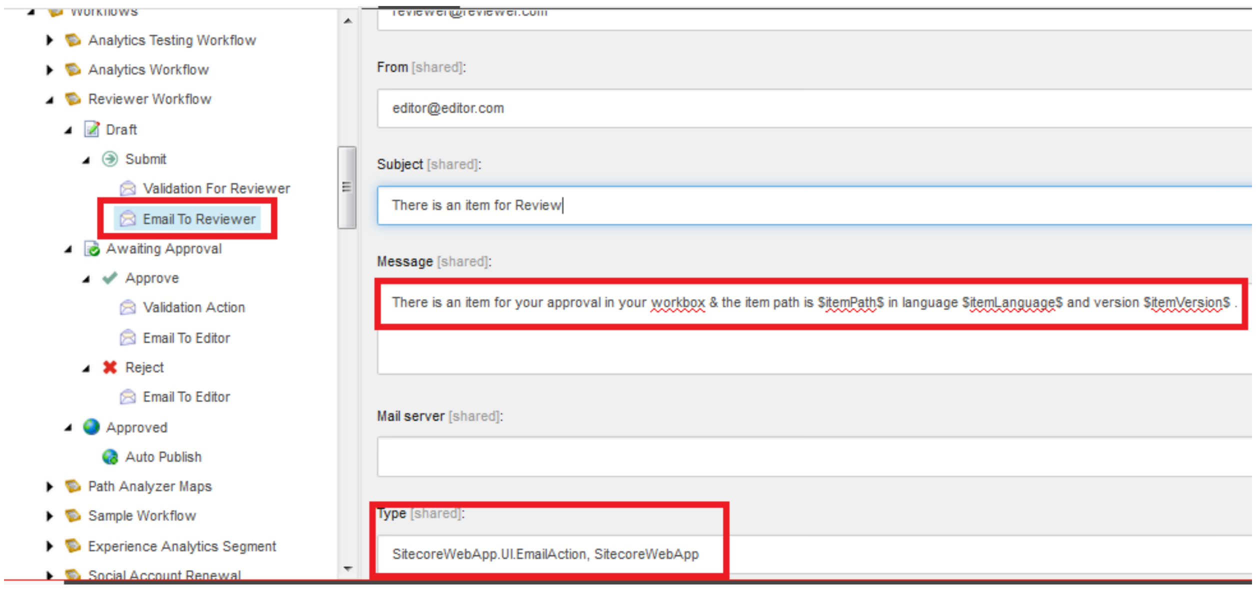


Domain	Local name	Full name	Comment
sitecore	Reviewer 1	sitecore\Reviewer 1	User
sitecore	Reviewer 2	sitecore\Reviewer 2	User

Role
sitecore\List Manager Editors
sitecore\Reviewer Group
sitecore\Sitecore Client Account Managing
sitecore\Sitecore Client Advanced Publishing
sitecore\Sitecore Client Authoring
sitecore\Sitecore Client Bucket Management
sitecore\Sitecore Client Configuring
sitecore\Sitecore Client Designing
sitecore\Sitecore Client Developing

Email To Reviewer

We want to send mail to Reviewer with message "*There is an item for your approval in your workbox & the item path is $itemPath$ in language $itemLanguage$ and version $itemVersion$.*"
Where *$itemPath$, $itemLanguage$* and *$itemVersion$* should be replaced by their value in code specified in "Type" field as



reviewer@reviewer.com

From [shared]:

editor@editor.com

Subject [shared]:

There is an item for Review

Message [shared]:

There is an item for your approval in your workbox & the item path is $itemPath$ in language $itemLanguage$ and version $itemVersion$.

Mail server [shared]:

Type [shared]:

SitecoreWebApp.UI.EmailAction, SitecoreWebApp

Include below code for sending mail functionality

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using Sitecore.Data.Items;
using Sitecore.Security.Accounts;
using System.Web.Security;
using Sitecore.Diagnostics;
using Sitecore.Data;
using Sitecore.Data.Fields;
using Sitecore.Workflows.Simple;
using System.Net.Mail;
using System.Net;
using Sitecore.Configuration;


namespace SitecoreWebApp.UI
{
    public class EmailAction
    {
        // Methods
        private string GetText(Item commandItem, string field, WorkflowPipelineArgs args)
        {
            string text = commandItem[field];
            if (text.Length > 0)
            {
                return this.ReplaceVariables(text, args);
            }
            return string.Empty;
        }

        public void Process(WorkflowPipelineArgs args)
        {
            Assert.ArgumentNotNull(args, "args");
            ProcessorItem processorItem = args.ProcessorItem;
            if (processorItem != null)
            {
                Item innerItem = processorItem.InnerItem;
                string fullPath = innerItem.Paths.FullPath;

                //string from = this.GetText(innerItem, "from", args);

                //If you want to know who triggered the mail otherwise above from code is
              fine
                 string from = string.IsNullOrEmpty(Sitecore.Context.User.Profile.Email)
              ? this.GetText(innerItem, "from", args) :
              Sitecore.Context.User.Profile.Email;

                string to = this.GetText(innerItem, "to", args);


                if (!string.IsNullOrEmpty(to))
                {
                    try
                    {


                        //Network Credentials
                        string host = Settings.GetSetting("MailServer");
                        var smtpClient = new SmtpClient(host);
                        smtpClient.Credentials = new
NetworkCredential(Settings.GetSetting("MailServerUserName"),
Settings.GetSetting("MailServerPassword"));
                        int port;
                        smtpClient.Port =
int.TryParse(Settings.GetSetting("MailServerPort"), out port) ? port : 25;

                        string mailSubject = this.GetText(innerItem, "subject", args);
                        string mailMessage = this.GetText(innerItem, "message", args);
                        Error.Assert(to.Length > 0, "The 'To' field is not specified in
the mail action item: " + fullPath);
                        Error.Assert(from.Length > 0, "The 'From' field is not specified
in the mail action item: " + fullPath);
                        Error.Assert(mailSubject.Length > 0, "The 'Subject' field is not
specified in the mail action item: " + fullPath);
                        Error.Assert(host.Length > 0, "The 'Mail server' field is not
specified in the mail action item: " + fullPath);

                        foreach (string tempTo in to.Split(new char[] { ';' }))
                        {
                            if (!string.IsNullOrEmpty(tempTo))
                            {
                                MailMessage message = new MailMessage(from, tempTo)
                                {
```

```
                            Subject = mailSubject,
                            Body = mailMessage
                        };

                        try
                        {
                            //Send mail
                            smtpClient.Send(message);
                        }
                        catch (Exception ex)
                        {
                            string err = ex.Message;
                        }
                    }
                }
            }
            catch (Exception ex)
            {
                string err = ex.Message;
            }

        }

    }
}

private string ReplaceVariables(string text, WorkflowPipelineArgs args)
{
    text = text.Replace("$itemPath$", args.DataItem.Paths.FullPath);
    text = text.Replace("$itemLanguage$", args.DataItem.Language.ToString());
    text = text.Replace("$itemVersion$", args.DataItem.Version.ToString());
    return text;
}
```