



Social Connected Module 3.0 for Sitecore CMS 7.5

Administrator's and Developer's Guide

A Quick Start Guide and Configuration Reference for Administrators and Developers

Table of Contents

Chapter 1	Introduction.....	4
1.1	Features	5
1.2	Architectural Requirements and Limitations	6
Chapter 2	Social Connector.....	7
2.1	Use Case.....	8
2.2	Configuration Overview.....	10
2.3	Configuring Access to Social Networks	11
2.3.1	Creating and Configuring a Facebook Web Application	11
	Permissions that Require a Facebook Review	13
2.3.2	Creating and Configuring a Twitter Web Application	15
2.3.3	Creating and Configuring a LinkedIn Web Application	18
2.3.4	Creating and Configuring a Google+ Web Application	20
2.4	Configuring Access to the Visitor's Profile Information	23
2.4.1	Web Application Access Request for Facebook, Twitter, LinkedIn and Google+.....	23
2.5	Placing Social Connector Controls on a Web Page.....	25
2.6	Getting a Contact's Klout Score.....	26
2.7	Mapping the Fields in the Visitor Profile.....	28
2.8	Synchronizing User Information.....	29
2.9	Disabling the Attachment of a Visitor's Social Network Accounts to Their User Profile	30
2.10	Detaching a Facebook Profile	31
Chapter 3	Posting Messages to Social Networks When a Goal Is Triggered	32
3.1	Configuration.....	33
3.1.1	Use Case.....	33
3.1.2	Configuration Actions	33
3.2	Creating a Message	34
3.3	Asking the Visitor if they Want Messages Posted Automatically	35
3.4	Placing Goal Context Information in the Message Body	36
Chapter 4	Posting Content Messages to Social Networks	37
4.1	Use Case.....	38
4.2	Configuration Overview.....	39
4.2.1	Generating URLs.....	39
4.3	Configuring Access to Social Networks	40
4.3.1	Facebook.....	40
4.3.2	Twitter.....	40
4.3.3	Social Marketer Messages	40
4.4	Creating a Network Account Item	41
	Deleting a Network Account Item.....	45
4.5	Creating a Social Marketer Account Item	47
	Deleting a Social Marketer Account Item	48
4.6	Customizing a Workflow.....	49
4.7	Creating a Message	50
4.7.1	Posting Messages Automatically when Publishing Content Items	52
4.8	Reviewing Messages	53
4.8.1	Default Workflows	53
	Social Message Workflow	53
	Social Marketer Message Workflow.....	54
4.9	Tracking Statistics on a Content Message.....	56
Chapter 5	Analytics Features.....	58
5.1	Tracking Campaign Statistics on a Content Message	59
5.2	Putting Share Buttons on the Web Page	62
5.3	Tracking Share Buttons Usage	63

- 5.4 Personalization Rules for the Social Connected Module 64
- 5.5 Tracking Visits from Social Networks..... 66
- Chapter 6 Developer's Notes 67
 - 6.1 Configuring the Module on Scaled Environment..... 68
 - 6.1.1 Multi-server Environment 68
 - Configuring the servers 68
 - 6.2 Working with Detached Messages..... 69
 - 6.3 The Social Connected Module Pipelines 70
 - 6.3.1 The social.CreateMessage Pipeline..... 70
 - 6.3.2 The social.ReadMessage Pipeline 70
 - 6.3.3 The social.PostMessage Pipeline 70
 - 6.3.4 The social.BuildMessage Pipeline 70
 - 6.3.5 The social.MatchUser Pipeline..... 71
 - 6.3.6 The social.InitializeApi Pipeline 71
 - 6.4 Configuring the Messages Search 72
 - 6.4.1 Solr Configuration..... 73
 - 6.5 The Social Connected Events..... 74
- Chapter 7 Tips and Tricks 75
 - 7.1 The Security Roles Used in the Social Connected Module 76
 - 7.2 How to Use the Error Handling Mechanism 77
 - 7.2.1 Error Handling on the Website 77
 - 7.3 Configuring Advanced Logging 78

Chapter 1

Introduction

This document is designed for Sitecore administrators and developers. It contains information about how to set up, configure, and tune the module.

The document contains the following chapters:

- **Chapter 1 — Introduction**
An introduction to the Social Connected module.
- **Chapter 2 – Social Connector**
Instructions for administrators about configuring and using the Social Connector.
- **Chapter 3 – Posting Messages to Social Networks When a Goal Is Triggered**
Instructions for administrators how to configure Goal Messaging.
- **Chapter 4 – Posting Content Messages to Social Networks**
Instructions for administrators how to configure Content Messaging.
- **Chapter 5 – Analytics Features**
A description of the analytics features in the module.
- **Chapter 6 – Developer’s Notes**
Information about multi-server environment configuration, module pipelines, and processors.
- **Chapter 7 – Tips and Tricks**
Tips and tricks for administrators.

1.1 Features

The Social Connected module provides the following features:

Social Connector

Social Connector lets website visitors log in to the website using the credentials from their social network accounts. The website then receives information about the visitor from their social network profile. You can also use the Social Connector to personalize the website.

Posting Messages to Social Networks when a Goal is Triggered

This feature works with the Social Connector and posts messages to the visitor's timeline in the social network every time they trigger a goal on the website.

Posting Content Messages to Social Networks

This feature posts messages about Sitecore content automatically to the timelines of the selected social network accounts when you publish Sitecore items. You can also post messages manually without publishing Sitecore items.

Reviewing Messages

You can place a message in a workflow. Workflows ensure that messages move through a predefined set of states before you can post them, usually intended to ensure that content receives the appropriate reviews and approvals before posting to a social network.

Analytics

You can place **Like** and **Tweet** buttons on a web page as Sitecore controls and then use Sitecore analytics to track the activity of website visitors.

Note

In this document, the phrase "post message" means to send a message to a social network and put it on the Facebook timeline or the Twitter timeline.

1.2 Architectural Requirements and Limitations

The Social Connected module cooperates with third party products and services. You should therefore pay special attention to the requirements and limitations related to your Sitecore solution.

Internet Access to Social Services

In a distributed Sitecore solution that is built on one or more environments, such as a Content Management server and one or more Content Delivery servers, make sure the servers have access to the social network services over the Internet.

Configuring Security

The Social Connector creates a new Sitecore user the first time a website visitor uses their social network credentials to log in to the website. If the Sitecore solution is built on one or more environments, such as a Content Management server and one or more Content Delivery servers, make sure you configure security according to the Sitecore recommendations. Security is an inseparable part of the Core database that must be shared between all the Content Management and Content Delivery servers if you want to keep security accounts synchronized. If the Core database cannot be shared between the servers, use database replication instead.

For more information about configuring security in multi-environment solutions, see the *Sitecore Scaling Guide*, the section *Isolating Instances within the CM and CD Environments* and the section *How to: Address Security Caching Clearing*.

Chapter 2

Social Connector

Social Connector is a feature of the Social Connected module that lets website visitors log in to the website with their social network credentials. Sitecore does not get access to the visitor's credentials; it uses the API to communicate with the social network.

When a website visitor logs in with their social network credentials, the Social Connected module receives all the information from the visitor's profile that the visitor has allowed the social network to share. The module saves social profile information to a Sitecore user profile in the *extranet* domain and to a contact profile. A user profile stores service information such as access token, application id and so on. A contact profile stores the rest of the information such as interests, sex, age and so on.

On the one hand, Social Connector lets website visitors log in to the website with their social network credentials and thereby avoid the registration process. On the other hand, Social Connector lets the website collect more information about a website visitor from their social network profile.

This chapter contains the following sections:

- Use Case
- Configuration Overview
- Configuring Access to Social Networks
- Configuring Access to the Visitor's Profile Information
- Placing Social Connector Controls on a Web Page
- Getting a Contact's Klout Score
- Mapping the Fields in the Visitor Profile
- Synchronizing User Information
- Disabling the Attachment of a Visitor's Social Network Accounts to Their User Profile
- Detaching a Facebook Profile

2.1 Use Case

This use case outlines how a website visitor uses their Facebook credentials to log in to a website.

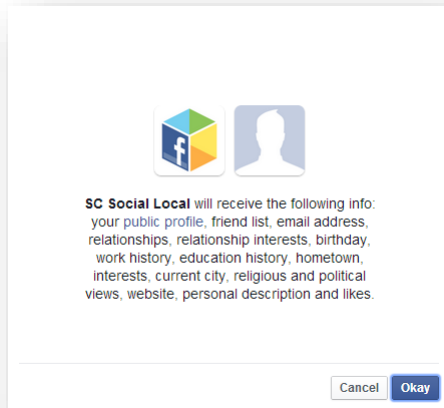
To connect with Facebook credentials:

1. A website visitor navigates to the website:

To log in to the website with their social network credentials, the visitor clicks the social network icon. The module redirects them to the social network login page.



2. On the login page, the visitor enters their credentials. The social network asks the visitor to grant the web application of the website access to the visitor's profile information.

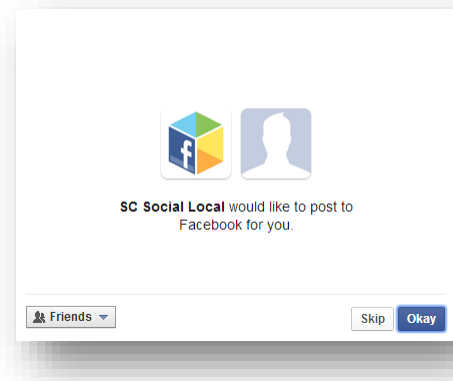


If the visitor does not grant access to their profile information, the system redirects them back to the website. The visitor still has to log in to the website.

Note

An administrator can configure the number of the fields that the web application requests access to. For more information about that, see the section *Configuring Access to the Visitor's Profile Information*.

If the visitor grants access to their profile information, the social network asks the visitor to grant the web application of the website access to post information to the social network.



Then the system redirects them back to the website where they are already logged in. Sitecore creates a new user profile or updates the information in the existing user profile.

3. The visitor can attach more social network accounts to their profile on the website. To attach more social network accounts, the visitor clicks the appropriate social network icon and grants access to the web application.

2.2 Configuration Overview

To configure Social Connector:

1. Create a web application for each social network and a corresponding Sitecore item. This web application will concatenate the module with the social network. For more information about configuring a web application, see the section *Configuring Access to Social Networks*.
2. Configure an access request for the information that the web application will receive from the website visitor's profile on the social network. For more information about configuring an access request, see the section *Configuring Access to the Visitor's Profile Information*.
3. Configure and place log in controls for the social networks on the web page. For more information about how to place log in controls, see the section *Placing Social Connector Controls on a Web Page*.
4. Reduce the time that the Social Connector spends to find an existing user when the visitor logs in to the website. Add the following string to the `web.config` file, to the `configuration/system.web/profile/properties` section:

```
<add type="System.String" name="SC_SocialNetworkId" />
```

After you have performed the obligatory configuration actions, you can:

- Arrange a Sitecore scheduling agent to periodically update the user profile with information from the social network.

For more information about how to arrange a scheduling agent, see the section *Synchronizing User Information*.

- Disable attaching the visitor's social network accounts to their user profile.

For more information about how to disable attaching visitor's social network accounts, see the section *Disabling the Attachment of a Visitor's Social Network Accounts to Their User Profile*.

2.3 Configuring Access to Social Networks

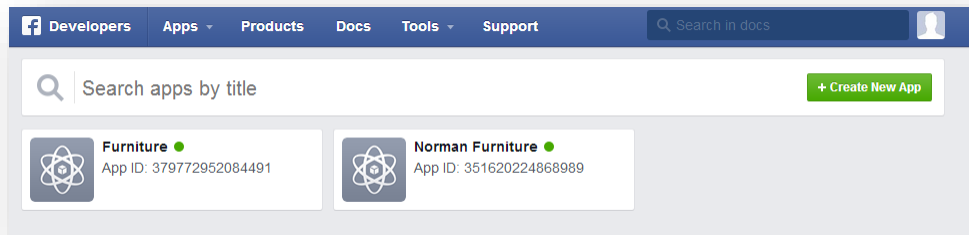
When interacting with social networks, Sitecore Social Connector uses the API of the social networks or the OAuth protocol for Google+. You must create a web application for every social network that you want the module to communicate with. You must also create a Sitecore item for each web application.

In a multisite scenario, you should create a web application for each website.

2.3.1 Creating and Configuring a Facebook Web Application

To create and configure a web application for Facebook:

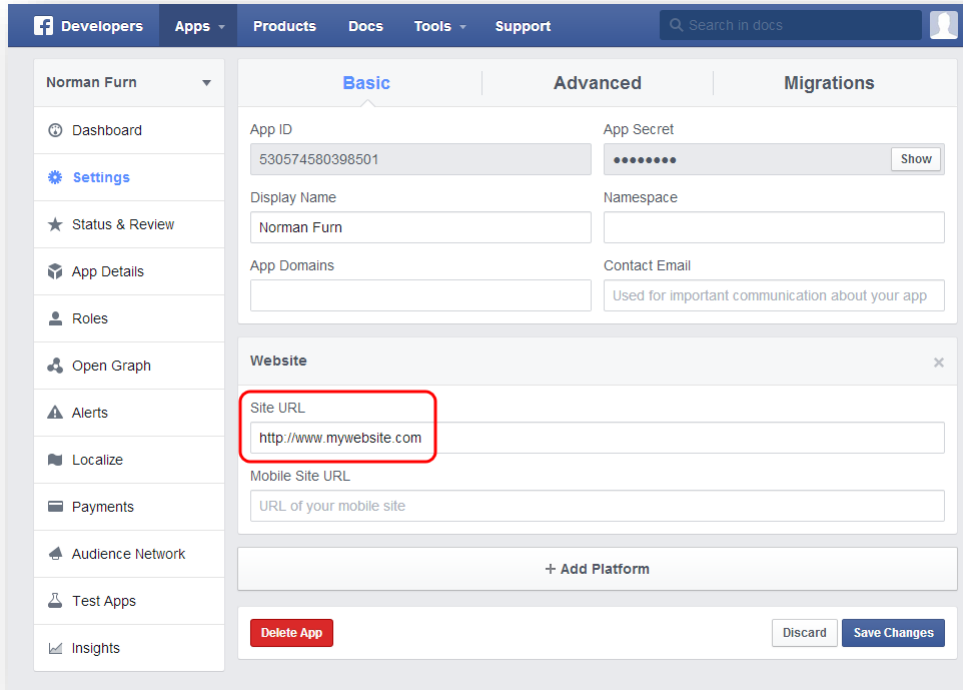
1. Go to <https://developers.facebook.com/apps> and create a new web application.



For more information about how to create a new web application, go to <https://developers.facebook.com/docs/web/tutorials/scrumptious/register-facebook-application/>

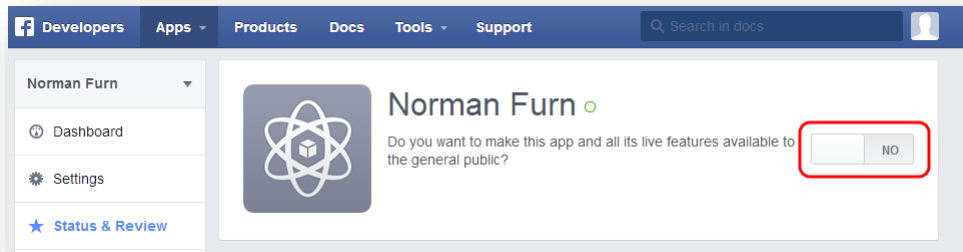
2. In the web application, on the **Settings** page, in the **Basic** tab, fill in all the required fields.

3. Add a **Website** platform, and in the **Site URL** field, enter the URL of the web page where you want to place the login controls.



The screenshot shows the Facebook Developer console interface. The left sidebar contains navigation options: Dashboard, Settings, Status & Review, App Details, Roles, Open Graph, Alerts, Localize, Payments, Audience Network, Test Apps, and Insights. The main content area is titled 'Basic' and includes fields for App ID (530574580398501), App Secret (masked with dots and a 'Show' button), Display Name (Norman Furn), Namespace, App Domains, and Contact Email (Used for important communication about your app). Below this is the 'Website' configuration section, which has a red box around the 'Site URL' field containing 'http://www.mywebsite.com'. Other fields include 'Mobile Site URL' and a '+ Add Platform' button. At the bottom, there are 'Delete App', 'Discard', and 'Save Changes' buttons.

4. On the **Status & Review** page, make your app publicly available.



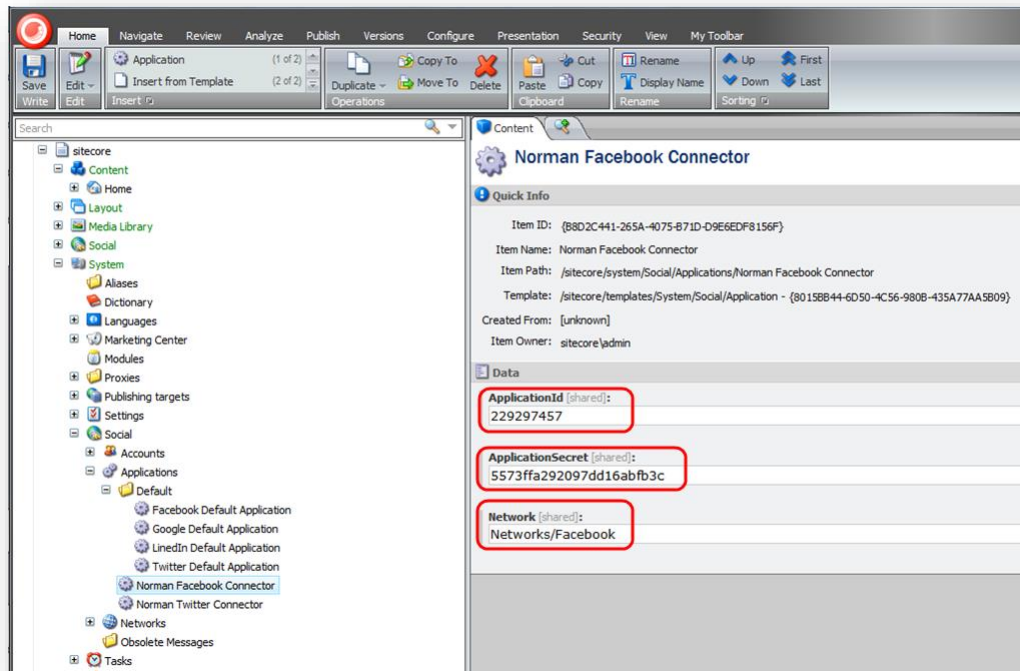
The screenshot shows the Facebook Developer console interface on the 'Status & Review' page. The left sidebar is the same as in the previous screenshot. The main content area shows the app name 'Norman Furn' with a green status indicator. Below the name is a question: 'Do you want to make this app and all its live features available to the general public?'. To the right of this question is a toggle switch, which is currently in the 'NO' position and is highlighted with a red box.

5. In Sitecore, in the **Content Editor**, navigate to the `sitecore/system/social/applications/default` folder.
6. In this folder, create an item based on the `sitecore/templates/system/social/application` template.

Important

The **Default** folder must contain only one item for the corresponding social network. If you are going to use several web applications for one social network, create the new items in the `sitecore/system/social/applications` folder.

7. In the new item, enter the appropriate information



Field	Value
ApplicationId	The App ID value of the web application
ApplicationSecret	The App Secret value of the web application
Network	Facebook

Permissions that Require a Facebook Review

Facebook reviews permissions that your application requests from your visitors.

However, Facebook does not review several permissions. The following permissions do not require Facebook approval:

Permission	Description
Public_profile	Gives access to a subset of a user's public profile fields: <ul style="list-style-type: none"> • id • name • first_name • last_name • link • gender • locale • age_range

Permission	Description
user_friends	This permission grants the application permission to read a list of the user's friends who also use your application. If any of your user's friends have in turn chosen not to share their list of friends with your application, the user's friends they will not show up in the list of friends for this user. Both people must have enabled the user_friends permission for a friend to show up in either friend list.
email	Provides access to the user's primary email address in the email property on the user object. Do not spam users. Your use of email must comply both with Facebook policies and with the CAN-SPAM Act.

Applications that work with the Social Connected module require Facebook approval for the following permissions:

Permission	Description
user_birthday	Used in Social Connector, personalization. Provides access to the birthday with the year as the birthday property. Note that your application may determine if a user is 'old enough' to use an application by obtaining the age_range public profile property.
user_education_history	Used in Social Connector, personalization. Provides access to education history as the education property.
user_interests	Used in Social Connector, personalization. Provides access to the user's list of interests as the interests connection.
user_likes	Used in Social Connector, personalization. Provides access to the list of all the pages that the user has liked as the likes connection.
user_location	Used in Social Connector, personalization. Provides access to the user's current city as the location property.
user_relationships	Used in Social Connector, personalization. Provides access to the user's family and personal relationships and relationship status.
user_religion_politics	Used in Social Connector, personalization. Provides access to the user's religious and political affiliations.
user_website	Used in Social Connector, personalization. Provides access to the user's web site URL.

Permission	Description
user_work_history	Used in Social Connector, personalization. Provides access to work history as the work property.
read_stream	Used to get statistics on a content message. Provides access to all the posts in the person's news feed. This permission is reserved for apps that replicate the Facebook client on platforms that don't have a native client. It may only be used to provide people with access to this content. It may take up to 14 days for your app to be reviewed.
publish_actions	Used in content posting and goal posting. Enables your app to post content, comments and likes to a user's stream and requires extra permissions from a person using your app. Because this permission lets you publish on behalf of a user please read the Platform Policies to ensure you understand how to properly use this permission. Note, you do not need to request thepublish_actions permission in order to use the Feed Dialog, theRequests Dialog or the Send Dialog. Facebook used to have a permission called publish_stream. publish_actions replaces it in all cases. This permission also replaces photo_upload.
manage_pages	Used in content posting. Enables your application to retrieve access_tokens for Pages and Applications that the user administrates. The access tokens can be queried by calling /{user_id}/accounts via the Graph API. See here for generating long-lived Page access tokens that do not expire after 60 days.

You can enable more fields that you will request from the visitor in the `Sitecore.Social.ProfileMapping.Facebook.config` file. Those fields require a Facebook review as well.

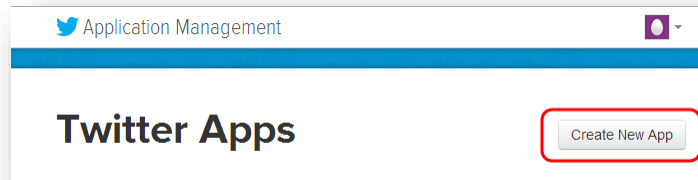
For more information about permissions with Facebook login, go to <https://developers.facebook.com/docs/facebook-login/permissions/v2.1>.

For more information about Facebook review guidelines, go to <https://developers.facebook.com/docs/apps/review>.

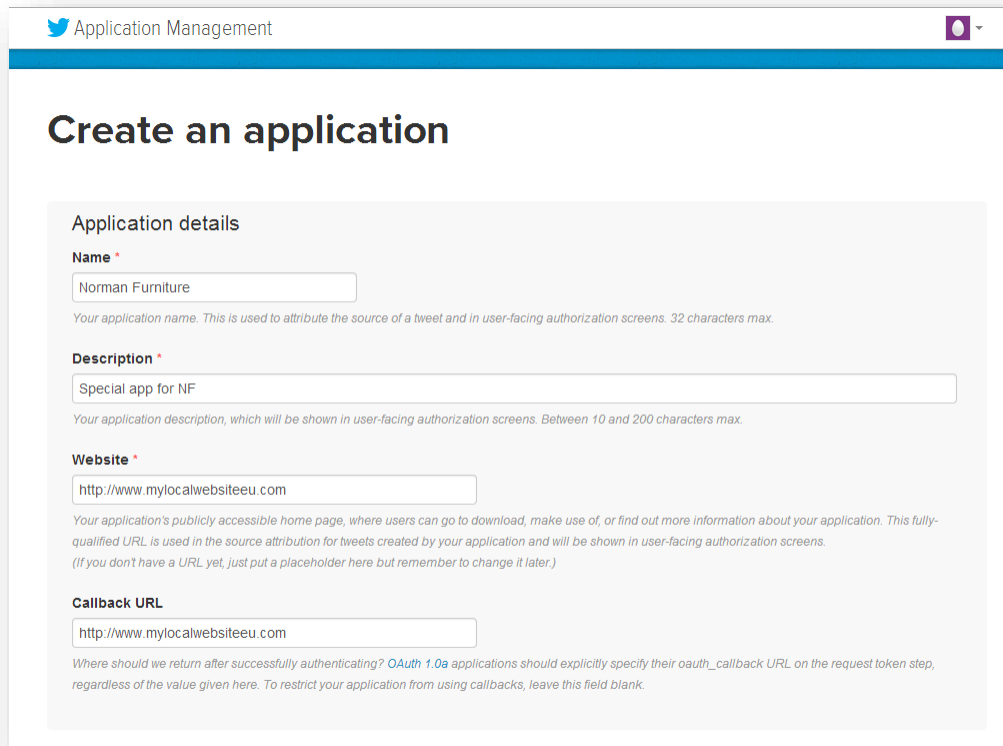
2.3.2 Creating and Configuring a Twitter Web Application

To create and configure a web application for Twitter:

1. Go to <https://dev.twitter.com/apps> and create a new web application.



2. In the web application, in the **Application details** section, fill in all the required fields.



Application details

Name *

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description *

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website *

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens. (If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL

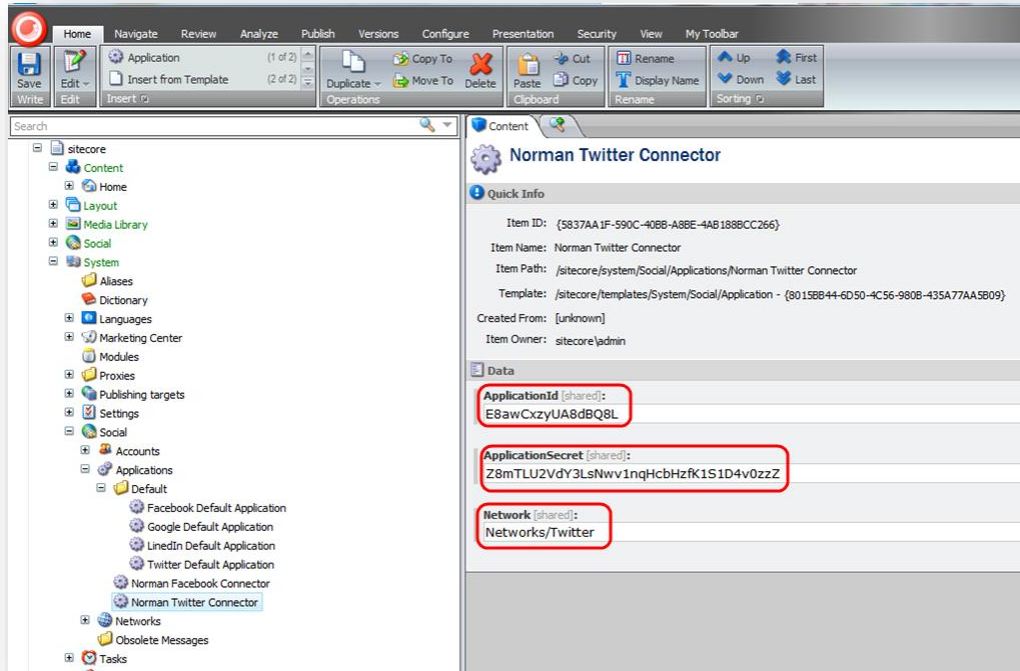
Where should we return after successfully authenticating? OAuth 1.0a applications should explicitly specify their oauth_callback URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.

3. In the **Callback URL** field, enter the URL of the webpage where you want to place the login controls.
4. Click **Create a Twitter application**.
5. After you created an application, on the **Permissions** tab, select **Read, Write and Access direct messages** and click **Update settings**.
6. In Sitecore, in the **Content Editor**, navigate to the `sitecore/system/social/applications/default` folder.
7. In this folder, create an item based on the `sitecore/templates/system/social/application` template.

Important

The **Default** folder must contain only one item for the corresponding social network. If you are going to use several web applications for one social network, create the new items in the `sitecore/system/social/applications` folder.

- In the new item, enter the appropriate information:

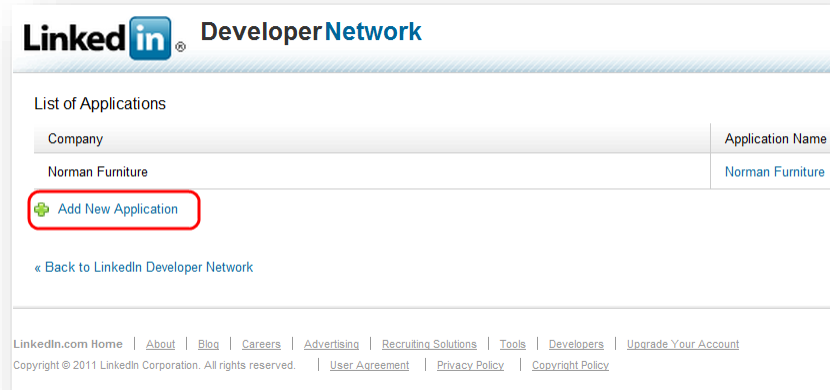


Field	Value
ApplicationId	The API Key value of the web application
ApplicationSecret	The API Secret value of the web application
Network	Twitter

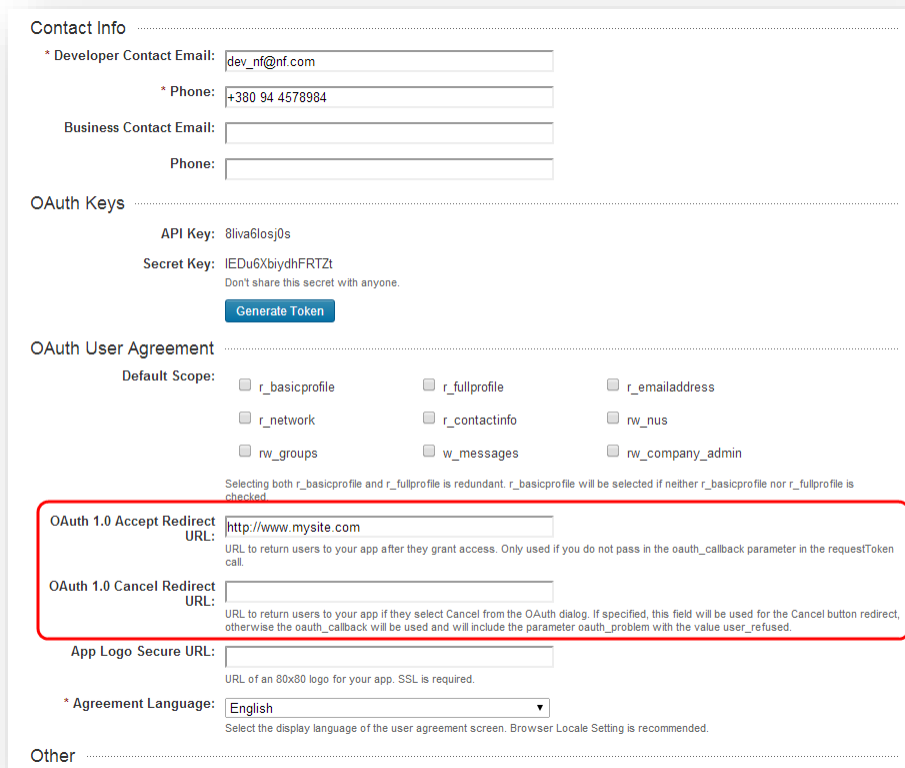
2.3.3 Creating and Configuring a LinkedIn Web Application

To create and configure a web application for LinkedIn:

1. Go to <https://www.linkedin.com/secure/developer> and create a new web application.



2. In the web application, fill in all the required fields.
3. In the **OAuth User Agreement** section, in the **OAuth 1.0 Accept Redirect URL** field, enter the URL of the web page where you want to place the login controls.



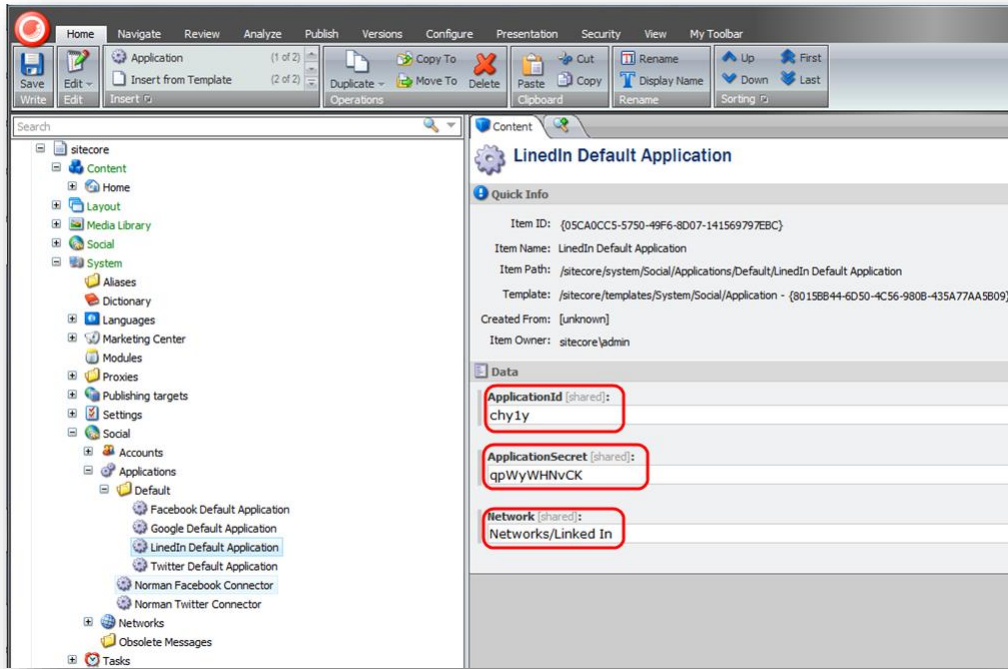
The **OAuth 1.0 Cancel Redirect URL** field must be blank.

4. In Sitecore, in the **Content Editor**, navigate to the `sitecore/system/social/applications/default` folder.
5. In this folder, create an item based on the `sitecore/templates/system/social/application` template.

Important

The `Default` folder must contain only one item for the corresponding social network. If you are going to use several web applications for one social network, create the new items in the `sitecore/system/social/applications` folder.

6. In the new item, enter the appropriate information:

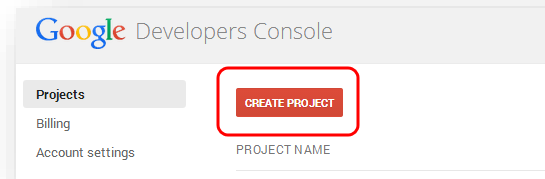


Field	Value
ApplicationId	The API Key value of the web application
ApplicationSecret	The Secret Key value of the web application
Network	LinkedIn

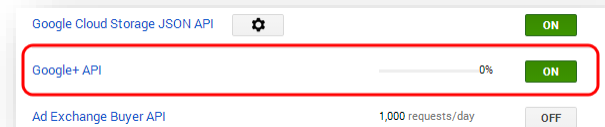
2.3.4 Creating and Configuring a Google+ Web Application

To create and configure a web application for Google+:

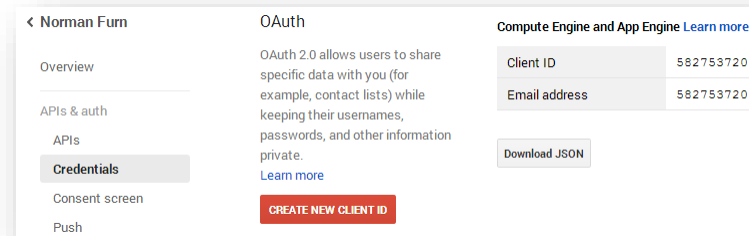
1. Go to <https://console.developers.google.com/project> and create a new project:



2. In the **Create Project** dialog box, enter the name of the project and then click **Create**.
3. On the **APIs & auth** page, set **Google+ API** to **On**.

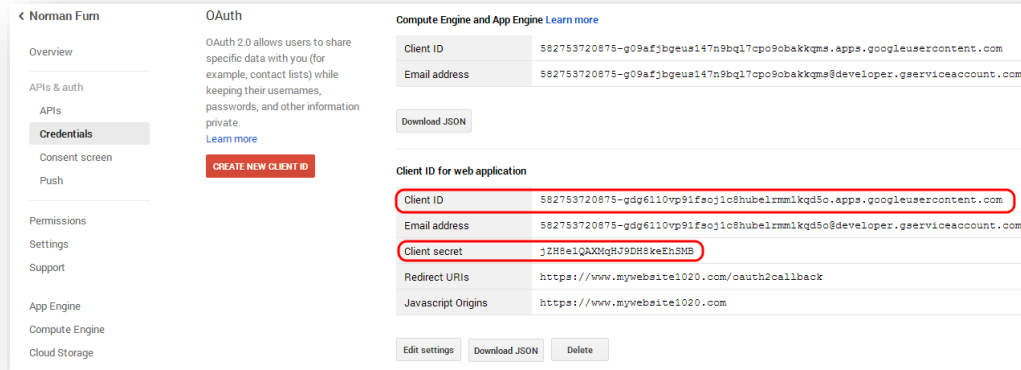


4. On the **Credentials** page, click **Create new client ID**.



5. In the **Create Client ID** dialog box, select **Web Application**.
6. In the **Create Client ID** dialog box, enter the following URLs in the **Authorized Javascript origins** field:
http://hostname
7. In the **Create Client ID** dialog box, enter the following URLs in the **Authorized redirect URI** field:
http://hostname/layouts/system/Social/Connector/SocialLogin.ashx?type=google_access
http://hostname/layouts/system/Social/Connector/SocialLogin.ashx?type=google_add
http://hostname/layouts/system/Social/Connector/SocialLogin.ashx?type=access
where *hostname* is the name of the host that is running the website.

- On the **API Access** page, save the values of the **Client ID** and **Client secret** fields.

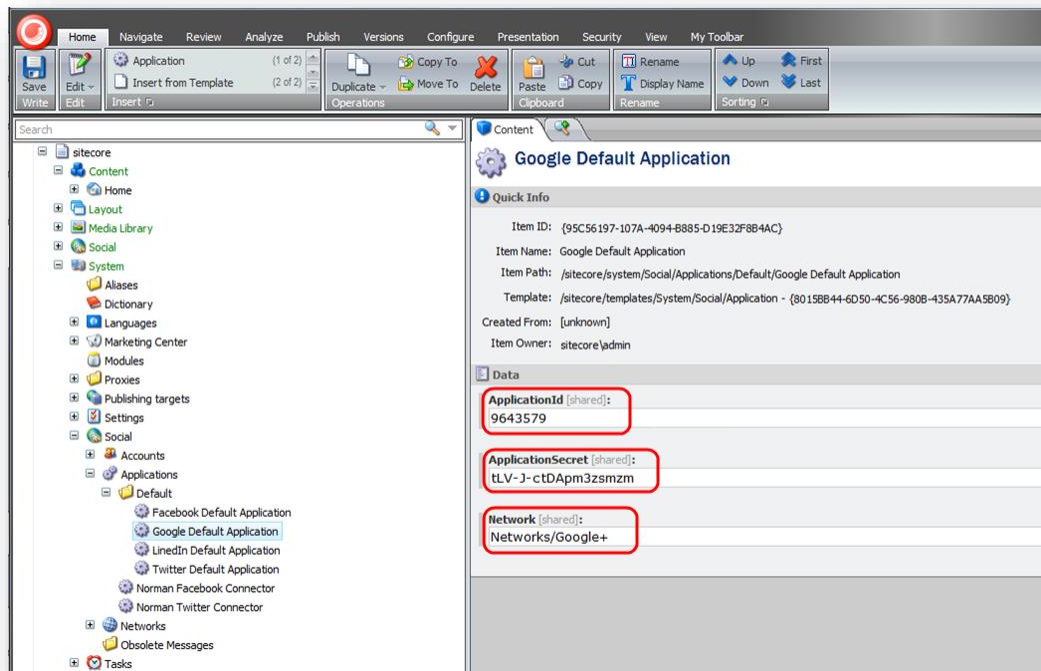


- In Sitecore, in the **Content Editor**, navigate to the `sitecore/system/social/applications/default` folder.
- In this folder, create an item based on the `sitecore/templates/system/social/application` template.

Important

The `Default` folder must contain only one item for the corresponding social network. If you are going to use several web applications for one social network, create the new items in the `sitecore/system/social/applications` folder.

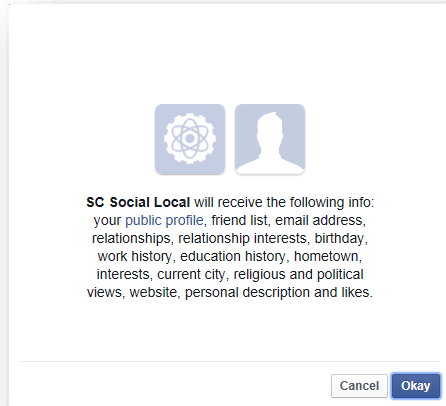
- In the new item, enter the appropriate information:



Field	Value
ApplicationId	The Client ID value of the web application
ApplicationSecret	The Client secret value of the web application
Network	Google+

2.4 Configuring Access to the Visitor's Profile Information

When a visitor logs in to the website with their social network credentials, the module redirects them to the login page of the social network. On that page, the visitor enters their credentials and the web application asks the visitor to grant it access to their profile information.



You can configure the exact information that the web application will ask for.

2.4.1 Web Application Access Request for Facebook, Twitter, LinkedIn and Google+

For Facebook, Twitter, LinkedIn, and Google+ you configure the access request in the `Sitecore.social.profilemapping.network_name.config` file.

To configure the fields that the web application can access:

1. Open the `[website_root]\app_config\include\social\sitecore.social.profilemapping.[network_name].config` file.
2. In the configuration section for the social network, in the configuration section for the field that you want the web application to use, set the `field enabled` parameter to `true`.
3. If the field is private, you must set its `permission` parameter according to the official social network documentation.

Important

We recommend that you do not disable the fields that are enabled by default because this might affect the presentation details in the module UI.

For example, to get access to the **Groups**' field on Facebook, the field section in the `Sitecore.social.profilemapping.Facebook.config` file should look like this:

```
<!-- 'Groups' field. -->
<!-- The Groups that the user belongs to. -->
<field enabled="true" originalKey="" sitecoreKey="fb_groups" permission="user_groups"
access="/me/groups" text="Groups" />
```

For more information about user profile fields, see the official documentation of the social network:

- For Facebook documentation, go to <http://developers.facebook.com/docs/reference/api/user/>.

- For Twitter documentation, go to <https://dev.twitter.com/docs/api/1.1/get/users/show>
- For LinkedIn documentation, go to <https://developer.linkedin.com/documents/profile-fields>.
- For new Google+ API, go to <https://developers.google.com/+/api/latest/people>.

2.5 Placing Social Connector Controls on a Web Page

You can configure and place log in controls for the social networks on the website. You can find the controls in the `sitecore/layout/sublayouts/social/connector` folder. For the MVC layout, use the Social Connected renderings in the `sitecore/layouts/Renderings/Social MVC/Connector` folder. You can fully customize these controls (change images, texts, and so on).

By default, the Social Connector controls use social web applications that are stored in the `sitecore/system/social/applications/default` folder. You can set another web application for the Social Connector control:

- In the UI:
 - 1) In the **Content Editor**, select the item presentation details
 - 2) Add a Social Connector control.
 - 3) Edit the control
 - 4) In the **Application** field, select the application.
- Using the API:
 - In the `ApplicationItemId` parameter, specify the ID of the Sitecore item containing social web application settings.

For a code sample, see the section *Disabling the Attachment of a Visitor's Social Network Accounts to Their User Profile*.

To set the webpage that the visitor will be redirected to after they have entered social network credentials and granted access to their profile information:

1. In the **Content Editor**, select the item presentation details.
2. Add a Social Connector control.
3. Edit the control.
4. In the **Call URL** field, enter the URL of the webpage.

2.6 Getting a Contact's Klout Score

The Social Connected module can access information about a contact's Klout Score if the contact logs in to the website with Twitter or Google+ credentials, or the contact attaches one or both of these social networks to their profile. The Social Connected module transfers their Klout Score to the Sitecore analytics database – Mongo DB.

To configure the Social Connected module to get a Klout Score:

Important

When you register your app on Klout's website, you enter into an agreement with Klout, who is an independent third party. Sitecore is not responsible for Klout's services or any losses arising from usage of these services.

1. In the `Website\app_config\include\social` folder, rename `Sitecore.Social.Klout.config.disabled` to `Sitecore.Social.Klout.config`.
2. Create an application on the Klout website. This web application links the module with the social networks.
3. Create an item for the application in Sitecore. This item contains application keys that the Social Connected module uses to cooperate with the application.
4. Configure a login control if needed. You can select a non-default application.

Creating an Application on the Klout Website

To create a Klout web application:

1. Go to <https://secure.mashery.com/login/developer.klout.com/>.
2. After logging in, click **Register to build your awesome app**.
3. On the **Applications** tab, click **Create a New Application**.
4. Fill in the required fields.

Creating an Item for the Application in Sitecore

To create an item for the Klout web application in Sitecore:

1. In Sitecore, in the **Content Editor**, create a new item in the `/sitecore/system/Social/Klout/Applications/Default` folder. The item must be based on the `/sitecore/templates/System/Social/Klout/Klout Application` template.
2. In the new item, enter the following information:

Field	Value
Application Id	The ID value of the web application. To get the ID, go to http://developer.klout.com/apps/myapps , click Edit application. In the URL, you can see the ID: http://developer.klout.com/apps/edit/ID .
Application Key	The Key value of the web application. You can get the key on this page: http://developer.klout.com/apps/mykeys .
Shared Secret	The Shared Secret value of the web application. You can get the value on this page: http://developer.klout.com/apps/mykeys

Configuring a Login Control

By default, Twitter and Google+ controls use the default Klout application that is stored in the `/sitecore/system/Social/Klout/Applications/Default` folder. To use a non-default application in the Twitter and Google+ login controls:

1. In Sitecore, in the **Content Editor**, select the item that contains the **Login with Twitter** or **Login with Google** login control.
2. In the Content Editor ribbon, on the **Presentation** tab, click **Details**.
3. In the **Layout details** dialog box, select the **Login with Twitter** or **Login with Google** control, and click **Edit**.
4. In the **Klout Application** field, select the application that you want the control to use.

2.7 Mapping the Fields in the Visitor Profile

When a website visitor logs in to the website with their social network credentials, the module receives their visitor profile information from the social network. The module saves social profile information to a Sitecore user profile in the *extranet* domain and to a contact profile. The module saves the information from the social network to MongoDB. The Social Connected module uses prefixes for social network fields to process user profile information. For example, `fb` is a prefix for Facebook fields. You can change the field mappings in the `sitecore.social.profilemapping.network_name.config` file.

To change the field mappings between a social network visitor profile and the user profile on the website:

1. Open the `[website_root]\app_config\include\social\sitecore.social.profilemapping.[network_name].config` file.
2. In the configuration section for the social network, set the following fields according to the official social network API documentation:

Field	Description
originalKey	All social networks use this field.
Permission	Facebook and LinkedIn uses this field.
Access	Facebook uses this field.
xmlPath	LinkedIn uses this field.
text	All social networks use this field.

3. In the configuration section for the social network, set the **sitecoreKey** field to the name of the Sitecore database field.

Important

We recommend that you do not change the **sitecoreKey** field values.

For more information about the user profile fields, see the official documentation of the social network:

- For Facebook documentation, go to <http://developers.facebook.com/docs/reference/api/user/>.
- For LinkedIn documentation, go to <https://developer.linkedin.com/documents/profile-fields>.
- For new Google+ API, go to <https://developers.google.com/+/api/latest/people>.
- For Twitter documentation, go to <https://dev.twitter.com/docs/api/1.1/get/users/show>.

For more information about the **originalKey** field for Twitter, see the properties of the **TwitterUser** class in the **TweetSharp** library.

For more information about the **originalKey** field for Google+, see the **WellKnownAttribute** class in the **DotNetOpenAuth** library.

2.8 Synchronizing User Information

When a website visitor logs in to the website, Sitecore Connector receives and saves all the user profile information from the social network that the visitor has allowed the network to share. The module saves social profile information to a Sitecore user profile in the *extranet* domain and to a contact profile.

You can configure whether or not the module should update a user's profile every time they log in. In the `Sitecore.social.config` file, use the following parameters:

- `Social.ProfileUpdating.Enabled` – enables or disables updating user profiles on log in;
- `Social.ProfileUpdating.DaysBeforeExpiration` – the number of days after which the module should update the profile of a user when they log in.

The module receives user profile information in two stages:

- When the website visitor enters their social network credentials, the module receives basic data from the social network: ID, email, application key, application secret, access token, and access token secret. The module uses this information to identify the visitor.
- When the visitor is identified, the module receives all the information from the user profile that the visitor has allowed to be shared. As a result, this information comes later than basic data. However, you can use the new parameter in the `LoginUser` and `AttachUser` methods to configure the module so that it receives all the user information at the same time.

For a sample code, see the section, *Disabling the Attachment of a Visitor's Social Network Accounts to Their User Profile*.

To regularly update user information with the user profile data from the social network, you can use a scheduling agent. This allows you to synchronize the user information in Sitecore with the information in the social network regardless of the visitor's activity on the website. However, this scheduling agent is influenced by the `Social.ProfileUpdating.Enabled` and `Social.ProfileUpdating.DaysBeforeExpiration` settings.

You can set and run the scheduling agent in the `sitecore.social.config` file. To do this, in the `[website_root]\app_config\include\social\sitecore.social.config` file, in the `scheduling` section, set the `interval` value:

```
<!-- Agent to synchronize data with social networks. -->
  <agent type="Sitecore.Social.Client.Tasks.SynchronizeSocialMediaInfo,
Sitecore.Social.Client" method="Run" interval="00:00:00" />
```

When the module synchronizes information, it follows these rules:

- If the field in the social network user profile has been updated, Sitecore updates this field in the user profile.
- If the field in the social network user profile has become blank, Sitecore clears this field in the user profile.

2.9 Disabling the Attachment of a Visitor's Social Network Accounts to Their User Profile

The Social Connected module lets the visitor who logged in to the website with their social network account add one more social network account to their user profile by default. In this case, the module does not create a new account but adds the information to an existing user profile.

You can disable this functionality by editing the following API:

```
protected void FacebookLoginButtonOneClick(object sender, ImageClickEventArgs e)
{
    const bool updateSocialProfileAsync = true;

    var connectorClientManager = new ConnectorClientManager();

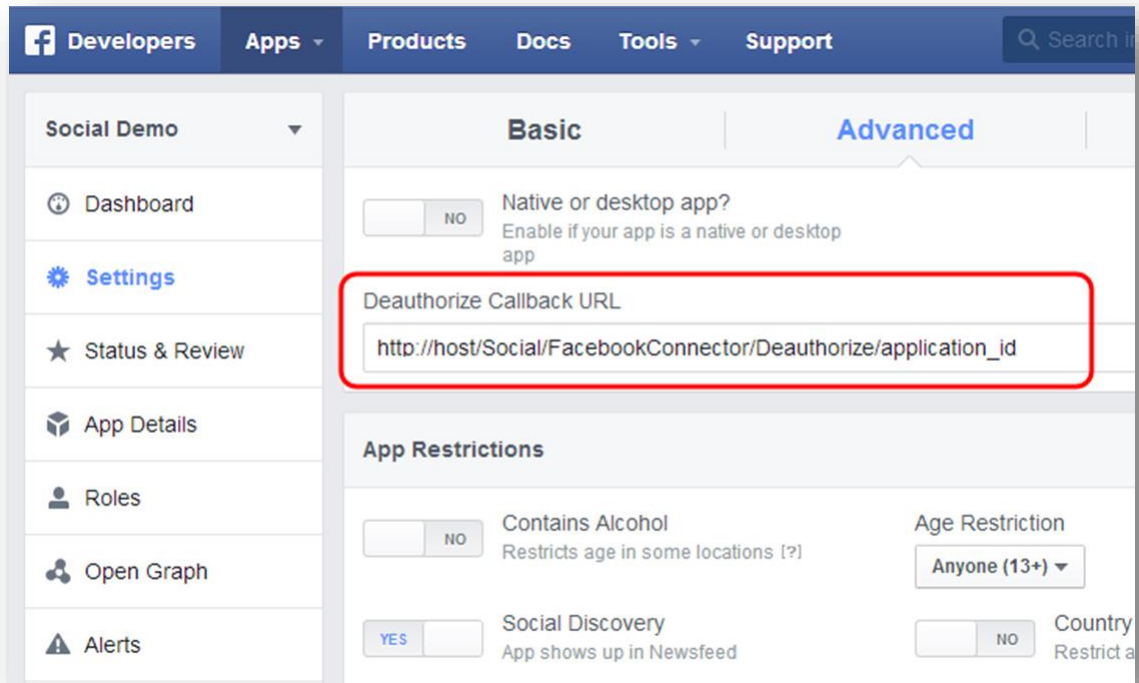
    if (!Sitecore.Context.User.IsAuthenticated)
    {
        connectorClientManager.LogOnUser(
            "Facebook",
            new ApplicationCredentials
            {
                ApplicationKey = "<your app key>",
                ApplicationSecret = "<your app secret>"
            },
            updateSocialProfileAsync);
    }
    else
    {
        connectorClientManager.AttachToUser(
            "Facebook",
            new ApplicationCredentials
            {
                ApplicationKey = "<your app key>",
                ApplicationSecret = "<your app secret>"
            },
            updateSocialProfileAsync);
    }
}
```

2.10 Detaching a Facebook Profile

To be able to log in to a website with their Facebook credentials, website visitors must allow the web application to get their user profile information from Facebook. If the visitor later forbids or deletes the application from their list of applications, you must not keep the Facebook information in the user profile.

To automatically remove the information in the user profile that was received from Facebook:

1. Navigate to the page of the Facebook application that you use for Social Connector.
2. Click **Settings**.



3. On the **Advanced** tab, in the **Deauthorize Callback URL** field, enter:
`http://<host>/Social/FacebookConnector/Deathorize/<application_id>`.

Chapter 3

Posting Messages to Social Networks When a Goal Is Triggered

The Social Connected module can automatically post messages to Facebook and Twitter when a website visitor triggers a goal on the website.

This feature requires that you configured Social Connector on the website.

For Facebook, you can post messages to the visitor's timeline. For Twitter, you can post tweets to the visitor timeline.

This chapter contains the following sections:

- Configuration
- Creating a Message
- Asking the Visitor if they Want Messages Posted Automatically
- Placing Goal Context Information in the Message Body

3.1 Configuration

The Social Connected module can post messages to social networks when a website visitor triggers a goal on the website. To enable this feature, you must configure the Social Connector because the module requires that the website visitor log in to the website with their social network credentials.

You can use this feature to post messages to Facebook and Twitter.

Before we explain how to configure posting messages on goal triggering, we will describe a use case in which the module automatically posts messages to social networks when a website visitor uploads a new image to the website.

3.1.1 Use Case

A visitor logs in to the website with their social network credentials. When the visitor triggers a goal, the module posts a message to the corresponding social network. You can configure the goal and the text used in the message.

Example:

A website visitor logs in to the website with their Facebook credentials. The visitor uploads an image *Friday Party* to the website. The *uploading an image* goal is triggered and the Social Connected module posts a message to the visitor's timeline on Facebook saying that the visitor uploaded the image called *Friday Party*.

3.1.2 Configuration Actions

To configure the posting of messages when a goal is achieved:

1. Configure Social Connector.

For more information about Social Connector configuration, see the section, *Configuration Overview*.

2. Create a goal.

For more information about configuring goals and page events, see the document [Marketing Operations Cookbook](#), chapter *Goals, Campaigns, and Events*.

3. Create a goal message that is sent to the social network.

For more information about editing messages, see the section *Creating a Message*.

After you have configured this, you can:

- Create a UI control that asks website visitors for permission to post messages to their social networks when they trigger goals.

For more information about creating UI control, see the section *Asking the Visitor if they Want Messages Posted Automatically*.

- Create personalized messages using goal context parameters.

For more information about the usage of goal context parameters, see the section *Placing Goal Context Information in the Message Body*.

Note

The analytics features used by the Social Connected module, such as goals, page events, and so on require Sitecore DMS 2.0.0 or later.

3.2 Creating a Message

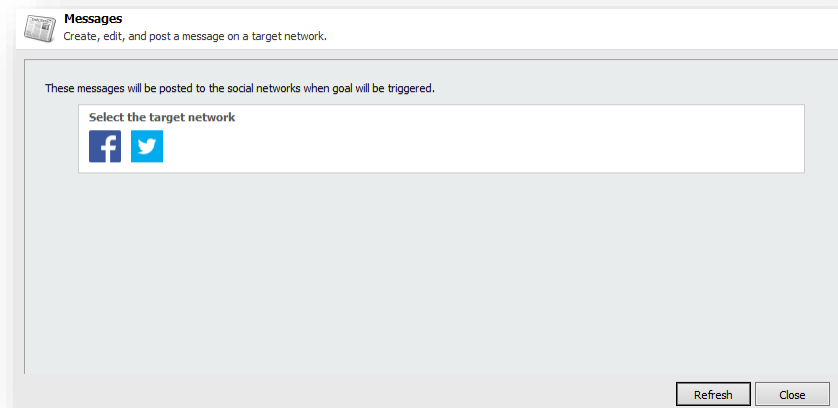
When you have configured Social Connector and assigned a goal to an item, you can edit the message.



Note

You can only post a message to a social network once.

To edit the message text:

1. In Sitecore, in the **Marketing Center**, select the goal that you want to bind a message to. Usually, the goals are stored in the folder `sitecore/system/marketing center/goals`.
2. In the **Marketing Center**, on the **Publish** tab, click **Messages**. The **Messages** dialog box appears.



3. In the **Messages** dialog box, click the **Create a Facebook Message** button  or the **Create a Twitter Message** button .
4. In the **Facebook message** or **Twitter message** dialog box, in the **Message** field, enter the text of the message. The rest of the fields are optional.

Note

Facebook limits the message text to 63000 symbols and Twitter limits the text to 140 symbols.

Note

If you select a campaign that will be triggered when a visitor clicks the link, you must deploy this campaign in advance.

5. In the **Facebook message** or the **Twitter message** dialog box, click **OK** to close the dialog box.

Important

You must publish the messages that you have created for the goals before they can be posted to the social networks. You can find the messages folder in the `Sitecore/Social/Messages` folder.

To put a URL in the middle of the message text, in the **Twitter message** dialog box, use the `$link` token in the **Message** field and enter the URL in the **Link** field. This token does not work for Facebook messages because Facebook structures the links according to Facebook's design.

3.3 Asking the Visitor if they Want Messages Posted Automatically

When you have configured Sitecore to post messages to the social network when a visitor achieves a goal on the website, this functionality applies to every visitor.

If you want to ask website visitors whether or not they want messages to be posted on their social network timelines, you must disable sending of messages by default and create a UI control to ask the visitor what they prefer:

1. In the `[website_root]\app_config\include\social\sitecore.social.config` file, set the value of the `Social.AllowMessageGoalPostingByDefault` setting to `false`.
2. Use the Sitecore API to create a UI control that asks the visitor whether or not they want to allow the Social Connected module to send messages to their social network timelines.

Use the following methods of the `Sitecore.Social.Api.PostingPreferenceManager` class:

- o `GetPostingPreference`
- o `SetPostingPreference`.

For example, the following code snippet asks the current visitor to decide whether or not to post a specific message. The message is specified in the message id:

```
var postingPreferenceManager = new PostingPreferenceManager();
var status = postingPreferenceManager.GetPostingPreference(Sitecore.Context.User,
messageId).AutomaticPostingPreference;
var newStatus = status == AutomaticPostingPreference.Allowed ?
AutomaticPostingPreference.Denied : AutomaticPostingPreference.Allowed;
postingPreferenceManager.SetPostingPreference(Sitecore.Context.User, messageId,
newStatus);
```

3.4 Placing Goal Context Information in the Message Body

You can use the Sitecore API to place some of the context information about how or when a goal is achieved in a personalized message that is posted on a social network.

For example, when a website visitor uploads an image to the website and triggers the *uploading image* goal, you can transfer the name of the image as a goal parameter and then add this information as a token to the message body. Use the

`Sitecore.Social.Client.Api.SocialPageEventManager` class and the `TriggerSocialPageEvent` method to transfer goal parameters.

For example, the following code snippet passes a username parameter to the social publishing engine:

```
var parameters = new Dictionary<string, string>
{
    {
        "username", Sitecore.Context.User.Name
    }
};
var socialPageEventManager = new SocialPageEventManager();
socialPageEventManager.TriggerSocialPageEvent("Login", parameters);
```

Chapter 4

Posting Content Messages to Social Networks

The Social Connected module can post messages to social networks when you publish Sitecore items. You bind the message to the content item, and the module posts this message automatically when you publish the content item. You can also post these messages manually without publishing the content items.

This feature works with Facebook and Twitter. You can post messages to a visitor's timeline on Facebook and to a Facebook page. You can post messages (tweets) to the timeline of a Twitter user who has used the web application to create a Twitter account.

The Social Connected module lets you create social marketer messages. A social marketer message is a request to a social media marketer. In this request, an author informs the social media marketer that new content is ready to be promoted in social networks. The media marketers must have access to Sitecore Workbox to be able to work with social marketer messages.

You can use the standard Sitecore workflow functionality to make sure the messages are reviewed before they are posted to social networks.

This chapter contains the following sections:

- Use Case
- Configuration Overview
- Configuring Access to Social Networks
- Creating a Network Account Item
- Creating a Social Marketer Account Item
- Customizing a Workflow
- Creating a Message
- Reviewing Messages
- Tracking Statistics on a Content Message

4.1 Use Case

This use case outlines how the module automatically posts messages to a social network when a new webpage is published on the website. All the messages are reviewed before they are posted to a social network.

Your company produces and sells smartphones. When the company produces a new smartphone, a new web page is created on the website describing the features. To attract more attention to the new model, you can automatically post messages to your website's timeline on Facebook or to the corresponding timeline on Twitter informing that you have unveiled a new product. This message can also contain the link to the new webpage. You can assign a campaign to this link, so you can view DMS statistics.

Your company has a set of accounts in various social networks:

- A Facebook fans page
- A Facebook official page
- A Twitter official timeline

In addition, your company has a media marketer who promotes the company products on social networks.

You control who creates, edits, approves, and posts social messages to social network accounts:

- Messages for the Facebook fans page are written and posted by content authors without reviewing.
- Messages for the official pages are written by content authors, reviewed by management, and posted by authors.
- Content authors create social marketer messages in which they ask media marketers to post messages to specific social networks.

This was an example of how you can use the Social Connected module to post content messages to social networks.

4.2 Configuration Overview

To configure the posting of content messages:

1. Create and configure a web application on the social network and create a Sitecore item for the web application. The web application links the module with the social network.
2. Create a Sitecore item for the social network account or the social marketer account.
3. Create a message and either post it immediately to the social network or post it later when you publish the Sitecore item.

To configure social marketer messages:

1. Create a Sitecore item for the social marketer account.
2. Create a social marketer message.

For more information about how to create and configure a web application, see the section *Configuring Access to Social Networks*.

For more information about how to create an account item, see the section *Creating a Network Account Item*.

For more information about how to create a social marketer account item, see the section *Creating a Social Marketer Account Item*.

For more information about how to implement a custom workflow for messages, see the section, *Customizing a Workflow*.

For more information about creating and posting messages, see the section, *Creating a Message*.

4.2.1 Generating URLs

In a single-site solution, where the Content Management and Content Delivery servers are on different domains, specify the content delivery URL in the `Sitecore.Social.config` file, in the `Social.LinkDomain` attribute.

In a multisite solution, on the Content Management server, in the `web.config` file, the module uses the path of the content item that the message is associated with to look for a matching website in the `<sites>` section and its `startItem` attribute. The module then uses the `hostName` attribute of that website to generate the URLs.

4.3 Configuring Access to Social Networks

When interacting with social networks, the module uses the API of the social networks. You must create a web application for every social network that you want the module to communicate with and create a Sitecore item for the social network account.

4.3.1 Facebook

To configure the posting of content messages to Facebook:

1. Create a new web application or use an existing one, and then create a Sitecore item for the web application.

For more information about creating a web application, see the section *Creating and Configuring a Facebook Web Application*.

2. In Sitecore, in the Content Editor, in the `/sitecore/system/Social/Accounts/` folder, create a new account item using the wizard. For more information about how to create a Facebook account item, see the section *Creating a Network Account Item*.

4.3.2 Twitter

To configure the posting of content messages to Twitter:

1. Create a new web application or use the existing one, and create a Sitecore item for the web application. For more information about how to create a web application, see the section *Creating and Configuring a Twitter Web Application*.

2. Set the required permissions on the web application.

For more information about the required permissions, see the section **Error! Reference source not found.**

3. In Sitecore, in the Content Editor, in the `/sitecore/system/Social/Accounts/` folder, use the wizard to create a new account item.

For more information about how to create a Twitter account item, see the section *Creating a Network Account Item*.

4.3.3 Social Marketer Messages

Content authors write a social marketer message and pass it to media marketers via workflow states. Media marketers post the message to social networks using external tools.

To configure social marketer messaging:

1. In Sitecore, in the Content Editor, in the `/sitecore/system/Social/Accounts/` folder, use the wizard to create a new social marketer account item.

For more information about how to create a social marketer account item, see the section *Creating a Social Marketer Account Item*.

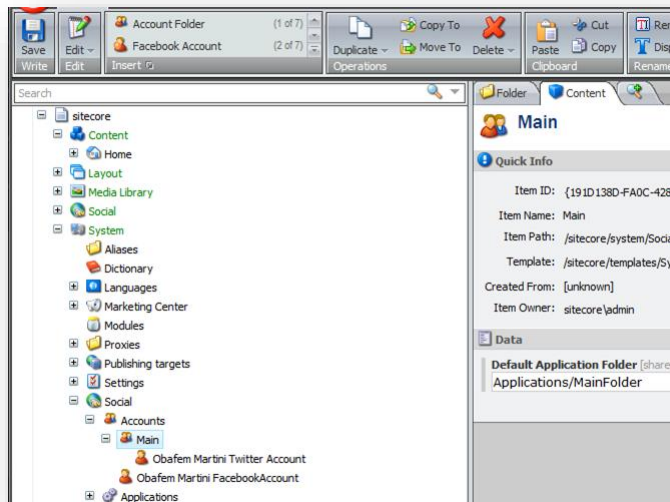
4.4 Creating a Network Account Item

Before you create an account item, you can create an account folder to store account items.

To create an account folder:

1. In the **Content Editor**, navigate to the `sitecore/system/social/accounts` folder.
2. Click the **Account Folder** insert option and enter the name of the folder.
3. In the new folder, on the **Content** tab, in the **Default Application Folder** field, specify the application folder.

The web applications in the application folder are used in the **Add a Social Network Account** wizard when an administrator creates accounts from the current account folder.



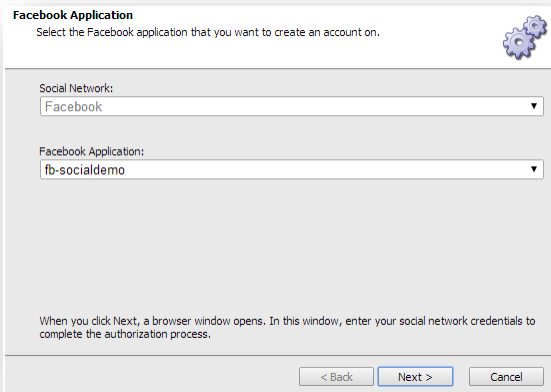
To create an account item:

1. In the **Content Editor**, navigate to the `sitecore/system/social/accounts` folder and select an account folder.
2. To start the wizard, click the **Facebook Account**, **Twitter Account**, **LinkedIn Account**, **Google+ Account** insert option.

The **Create a Social Network Account** wizard guides you through the process of creating a network account.

We will describe this process using the **Create a Facebook Network Account** wizard.

Facebook Application Page



Facebook Application
Select the Facebook application that you want to create an account on.

Social Network:
Facebook

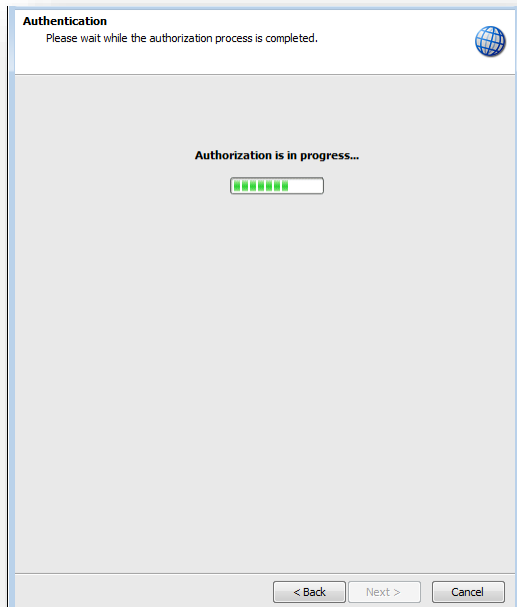
Facebook Application:
fb-socialdemo

When you click Next, a browser window opens. In this window, enter your social network credentials to complete the authorization process.

< Back Next > Cancel

In the **Facebook Application** field, select the web application that the account should use to cooperate with the social network.

Authentication Page



Authentication
Please wait while the authorization process is completed.

Authorization is in progress...

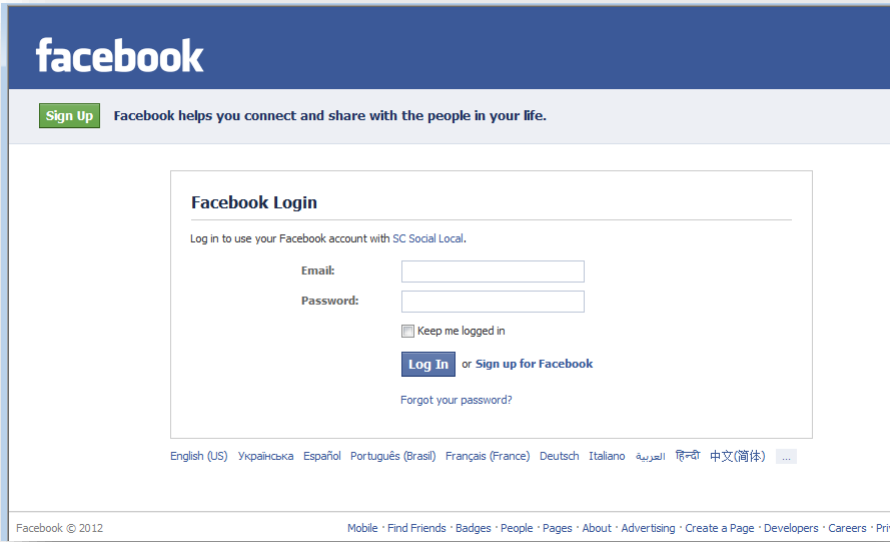
Progress bar: [|||||]

< Back Next > Cancel

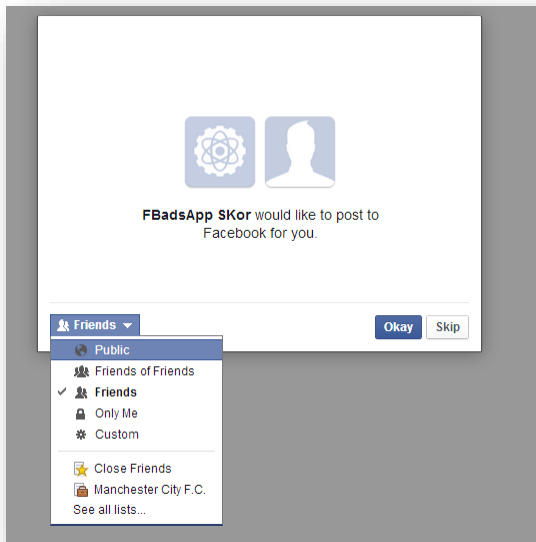
Note

An authentication session timeout is set to 60 seconds by default. During this period, the user can log in to the social network. If the user does not log in to a social network within this timeframe, the wizard stops. You can change this default timeout parameter in the `social.config` file, in the `Social.AddNetworkAccountWizard.AuthenticationTimeout` setting.

As a part of this step, the wizard opens a new browser window, in which you must enter the credentials of the social network user account. The module posts messages to the timeline of this user or its pages.



To define the user groups that can see the activity from the web application on Facebook, click the drop-down list and select one of the user groups.

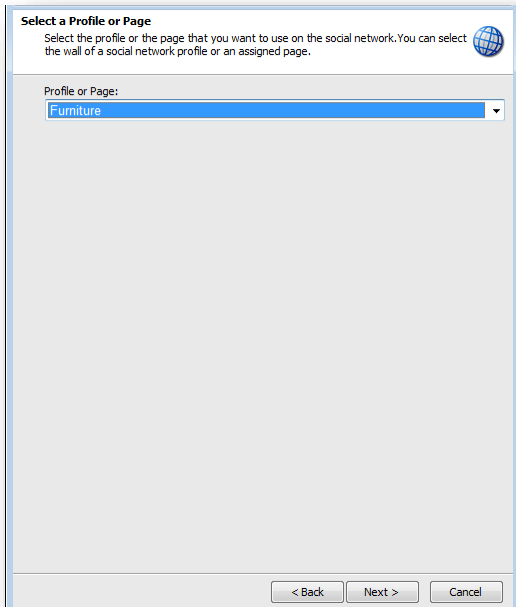


Note

To create an account, you must set all the required permissions to *Allow*.

Select a Profile or Page

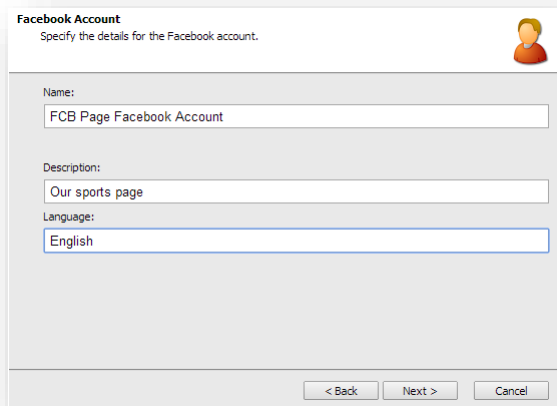
On the **Select a Profile or Page** page, you specify the profile or page that you want the module to update.



The module can post messages to the user's timeline or to any assigned Facebook pages. In the drop-down list, select where you want to post messages to.

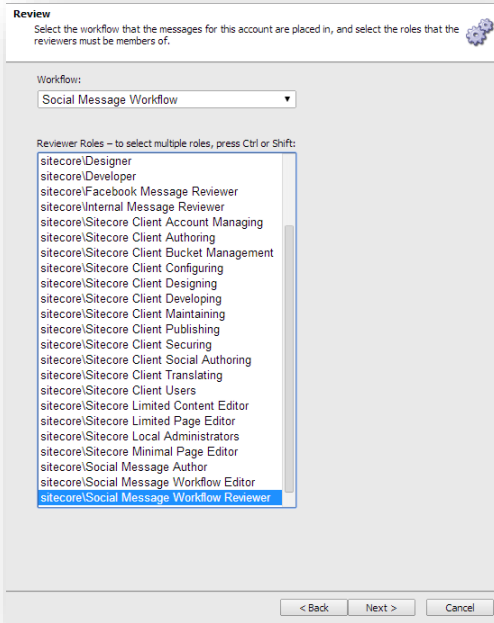
Facebook Account

On the **Facebook Account** page, enter the name for this new network account. You can also provide additional information about the account in the **Description** and **Language** fields.



Review Page

On the **Review** page, select a workflow to place the messages for the account in and assign roles for the reviewer.



For more information about security roles that you must assign to a reviewer, see the section, *Reviewing Messages*.

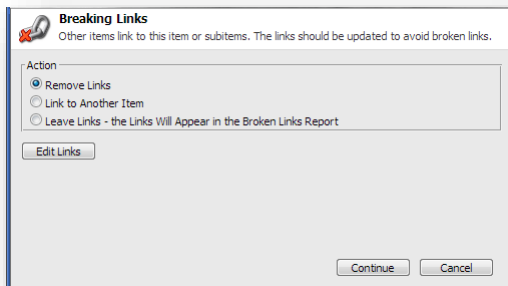
Final Page

On the last page of the wizard, click **Finish** to close it.

A new account item is added to the **Content Tree**.

Deleting a Network Account Item

You can delete a network account item as a standard Sitecore item. When you click **Delete the item** button, the **Breaking Links** dialog box appears:



If the item that you want to delete has attached messages, those messages are not deleted. However, the **Link to another item** option does not link the existing messages to another item. You must create a new message and move the content from the existing message to the new one.

4.5 Creating a Social Marketer Account Item

Before you create an account item, you can create an account folder to store account items.

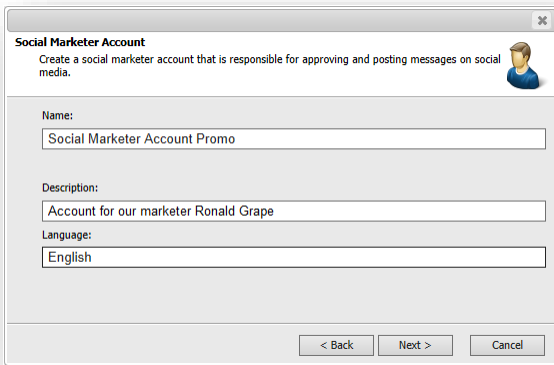
To create an account item:

1. In the **Content Editor**, navigate to the `sitecore/system/social/accounts` folder and select an account folder.
2. To start the **Create a Social Marketer Account** wizard, click the **Social Marketer Account** insert option.

The **Create a Social Marketer Account** wizard guides you through the process of creating an account.

Social Marketer Account Page

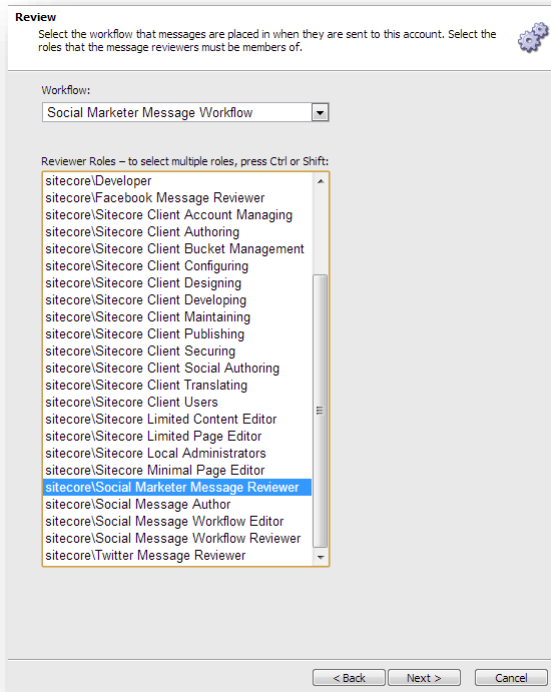
On the **Social Marketer Account** page, enter the name for this account. You can also provide additional information about the account in the **Description** and **Language** fields.



The screenshot shows a dialog box titled "Social Marketer Account" with a close button (X) in the top right corner. Below the title is a small icon of a person. The main text reads: "Create a social marketer account that is responsible for approving and posting messages on social media." There are three text input fields: "Name:" with the value "Social Marketer Account Promo", "Description:" with the value "Account for our marketer Ronald Grape", and "Language:" with the value "English". At the bottom, there are three buttons: "< Back", "Next >", and "Cancel".

Review Page

On the **Review** page, select a workflow to place the messages for the account in and assign roles for the reviewer.



For more information about security roles that you must assign to a reviewer, see the section, *Reviewing Messages*.

Final Page

On the last page of the wizard, click **Finish** to close it.

A new account item is added to the **Content Tree**.

Deleting a Social Marketer Account Item

You can delete a social marketer account item as a standard Sitecore item. For more information about deleting options, see the section *Deleting a Network Account Item*.

4.6 Customizing a Workflow

The Social Connected module provides a Social Message workflow that contains two states: *Draft* and *Awaiting Approval*. You can create your own workflow and place application messages into it.

To enable a custom workflow:

1. Create a workflow.

For more information about working with a workflow, see the *Workflow Cookbook* on the [Sitecore Developers Network](#).

2. Create security roles that have access to the workflow states and commands.
3. If you have several reviewers responsible for different social networks accounts, create a security role for each reviewer.
4. Create a social network account. In the **Create a Network Account** wizard, specify the new workflow and the roles for the reviewers.
5. Create users for authors and reviewers.

A user that is assigned to an author must be a member of the following security roles:

- sitecore\Author
- sitecore\Social Message Author
- The role that you created in step 2 of this procedure. This role grants access to the appropriate states of the workflow.

A user that is assigned to a reviewer must be a member of the following security roles:

- sitecore\Author
- The role that you created in step 3 of this procedure — it grants access to the messages of the corresponding social network account.
- The role that you created in step 2 of this procedure — it grants access to the workflow states in which the messages wait for approval.

For more information about security roles, see the section *The Security Roles Used in the Social Connected Module*.

4.7 Creating a Message

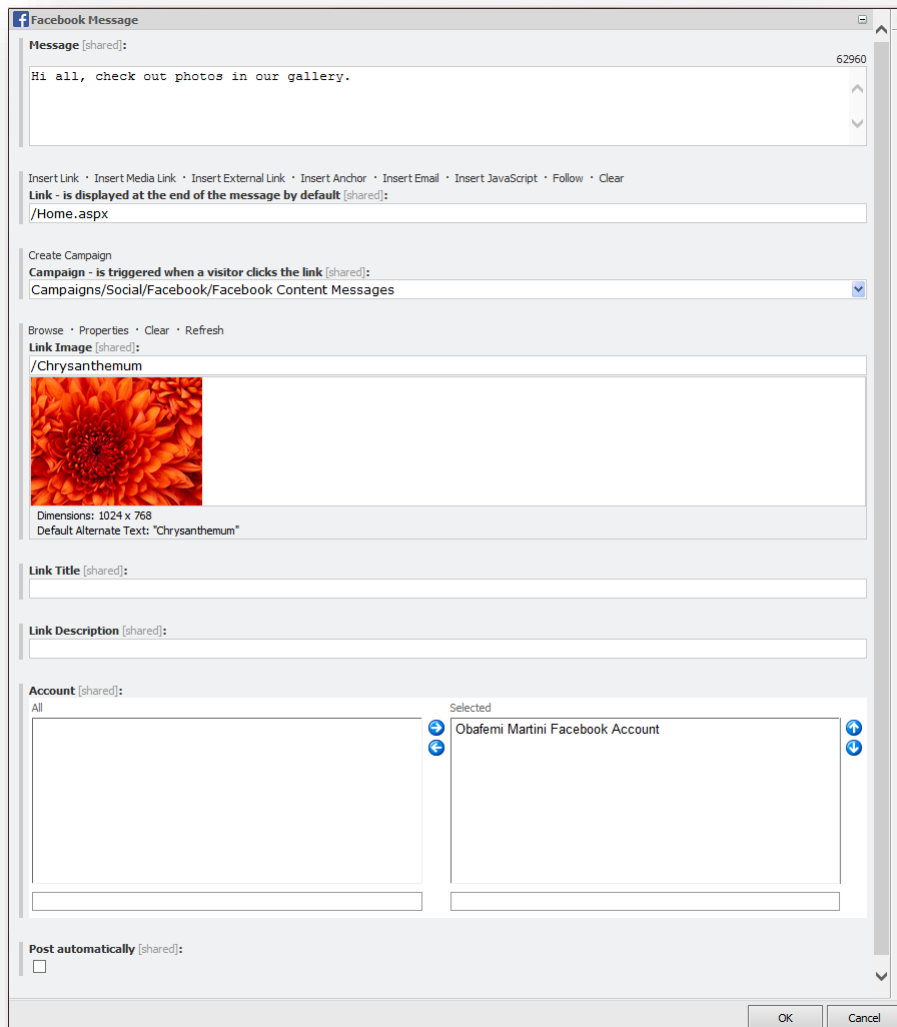
When you have configured the web application and created a Sitecore item for the social network or the social marketer account, you can create a message.

To create and post a message:

1. In the **Sitecore Desktop**, in the **Content Editor**, select the item that you want to bind a message to.
2. On the **Publish** tab, click **Messages**. The **Messages** dialog box appears.

Alternatively, you can create a message in the **Page Editor**. To open the **Messages** dialog box, in the **Page Editor**, on the **Home** tab, click **Messages**.

3. In the **Messages** dialog box, click the **Create a Facebook Message** button  or the **Create a Twitter Message** button  or the **Create a Social Marketer Message** button .



4. In the **Facebook Message**, **Twitter Message** or **Social Marketer Message** dialog box, in the **Message** field, enter the text of the message.

5. In the **Link** field, enter the link to the webpage that you want the website visitor to go to. By default, the link refers to the current page.
6. In the **Campaign** field, select the campaign that is triggered when a visitor clicks the link.
If you assign a campaign to the message, you will be able to review DMS message statistics including value, visits, and value per visits for all the visitors who click on a link in this message. Note that the campaign must be deployed before you can use it in the message.
7. Click **Create Campaign** to create a new campaign. The new campaign is stored in the appropriate folders under Marketing Center/Campaigns/Social.
8. To stylize the link, use the **Link Image**, **Link Title**, and **Link Description** fields. This is available for a Facebook message only.
9. To post the message automatically when you publish the current Sitecore item, select **Post automatically**. You cannot use this option for social marketer messages.

Note

If you use a workflow for account messages, only the messages in the final workflow state can be posted to a social network.

10. For a social marketer message, you can enter a comment to a media marketer in the **Comments** field.
11. In the **Account** field, select the social network or social marketer accounts that you want to post the message to. If you select several accounts, several messages are created – one message for every selected account.
12. Click **OK** to close the dialog box.
13. In the **Messages** dialog box, click **Post now** to post the message immediately.



In a Twitter message, the link is inserted at the end of the message by default. To put the URL in the middle of the message text, in the **Twitter message** dialog box, in the **Text** field, insert the `$link` token and in the **Link** field, enter the URL. This token does not work for Facebook messages because Facebook structures the links according to Facebook's design.

Note

You can only post a message to a social network once.

4.7.1 Posting Messages Automatically when Publishing Content Items

The Social Connected module uses a scheduling agent called `Sitecore.Social.Client.Tasks.PostFlaggedMessages` to post content messages automatically after you publish the Sitecore items that are related to the messages.

You can set and run the scheduling agent in the `sitecore.social.config` file. To do this, in the `[website_root]\app_config\include\social\sitecore.social.config` file, in the scheduling section, set the interval value:

```
<!-- Agent to post messages, marked to be posted automatically. -->
  <agent type="Sitecore.Social.Client.Tasks.PostFlaggedMessages,
Sitecore.Social.Client" method="Run" interval="01:00:00">
    <param desc="Social accounts root (item path or
ID) ">/sitecore/system/social/accounts</param>
  </agent>
```

This agent is ready to use, and it is configured to post every hour. Therefore, if you publish a Sitecore item, the corresponding message is posted next time the scheduling agent runs. As a result, there is a delay between when an item is published and when the corresponding message is posted.

Note

If you use a workflow for account messages, only the messages in the final workflow state can be posted to a social network.

4.8 Reviewing Messages

You assign a workflow to all the messages for the specific account when you create an account item.

If a message is in a workflow, it must be reviewed before you can post it to a social network.

A reviewer uses the **Workbox** – standard Sitecore tool to process items in workflows.

If you use default Social Connected workflows, you must assign the following security roles to a reviewer so that they can review the messages:

- sitecore\Author
- sitecore\Social Message Workflow Reviewer

For more information about the security roles that you must assign to a reviewer if you use a custom message workflow, see the section *Customizing a Workflow*.

For more information about security roles, see the section *The Security Roles Used in the Social Connected Module*.

To grant reviewers access to messages for different social networks, assign one of the following roles to a reviewer:

- sitecore\Facebook Message Reviewer
- sitecore\Twitter Message Reviewer
- sitecore\Social Marketer Message Reviewer

We recommend that you use the following combination of workflows and roles for accounts:

Social Network Account	Workflow	Reviewer Role
Facebook	Social Message Workflow	sitecore\Facebook Message Reviewer
Twitter	Social Message Workflow	sitecore\Twitter Message Reviewer
Social Marketer	Social Marketer Message Workflow	sitecore\Social Marketer Message Reviewer

For more information about all security roles that are used in the Social Connected module, see the section *The Security Roles Used in the Social Connected Module*.

4.8.1 Default Workflows

The Social Connected module provides two workflows by default:

- The *Social Message* workflow
This workflow is for messages that are posted to social networks.
- The *Social Marketer Message* workflow
This workflow is for social marketer messages to deliver them to media marketers.

Social Message Workflow

The *Social Message* workflow is designed for messages that will be posted to social networks. It contains three states:

State	Description
<i>Draft</i>	When a content author creates a message, the message is assigned to the <i>Draft</i> state.
<i>Awaiting Approval</i>	When the content author believes that the message is ready to be posted to a social network, they submit it to the <i>Awaiting Approval</i> state. A reviewer either submits the message to the <i>Approved</i> state or rejects it and moves it back to the <i>Draft</i> state.
<i>Approved</i>	When the message enters the <i>Approved</i> state, it can be posted to a social network automatically along with the content item or manually by clicking Post now .

Note

Only the messages in the *Approved* state can be posted to social networks.

The schema of the *Social Message* workflow is:

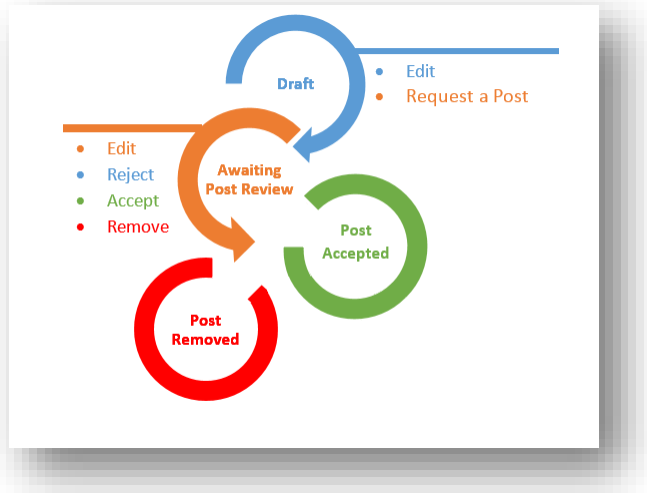
**Social Marketer Message Workflow**

The *Social Marketer Message* workflow is designed to pass a message to the media marketer who posts the message to social networks using external tools. The *Social Marketer Message* workflow contains four states:

State	Description
<i>Draft</i>	When a content author creates a message, the message is assigned to the <i>Draft</i> state.
<i>Awaiting Post Review</i>	When the content author believes that the message is ready to be posted to a social network, they submit it to the <i>Awaiting Post Review</i> state. A reviewer, usually a media marketer, submits the message to the <i>Post Accepted</i> state, rejects the message and moves it back to the <i>Draft</i> state or decides that the message does not fit a social network format moving it to the <i>Post Removed</i> state.

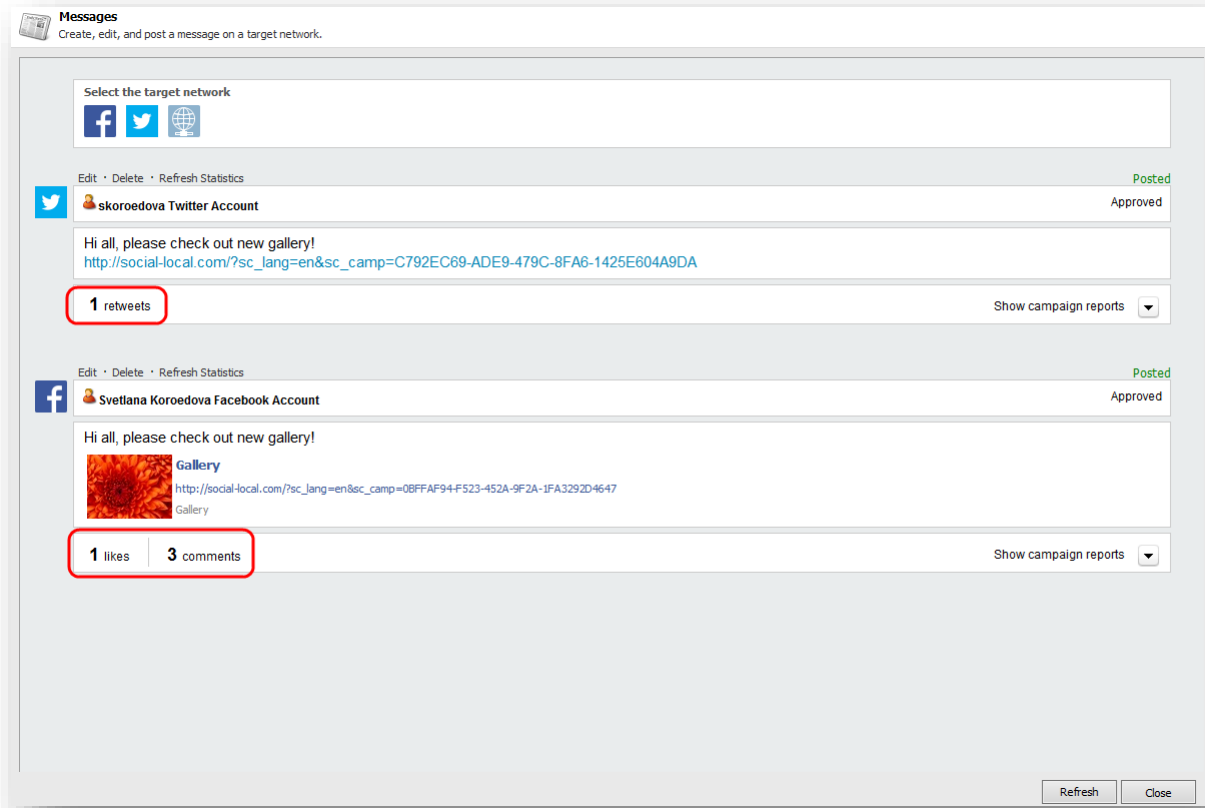
State	Description
<i>Post Removed</i>	The message in this state is never posted to a social network by the reviewer.
<i>Post Accepted</i>	The reviewer posts the message in this state to a social network.

The schema of the *Social Marketer Message* workflow is:



4.9 Tracking Statistics on a Content Message

On a content message, you can track the social media statistics: how many times the message is liked, commented, or retweeted.

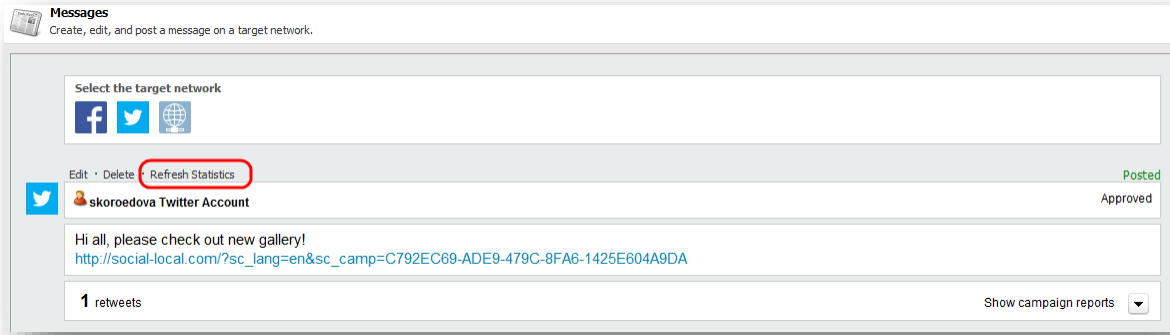


If you are using the Social Connected module with DMS, you can also see information about any campaigns that are associated with messages. For more information about campaign statistics, see the section *Tracking Campaign Statistics on a Content Message*.

You can update the social media statistics manually and automatically.

Manually Updating the Statistics

To update the message statistics manually, click **Refresh Statistics**.



The module uses APIs of respective social networks to retrieve the updated data.

Automatically Updating the Statistics

The Social Connected module uses the scheduling agent that is called `Sitecore.Social.Client.Tasks.UpdateMessagesStatistics` to update statistics. This agent is ready to use. The agent updates statistics at increasing time intervals depending on the *age* of the message, to balance the freshness of the data with the number of web requests to the social network APIs.

The module stops updating the statistics when the message is 107 days old. You can change this parameter in the `Sitecore.Social.config` file, in the `Social.FibonacciRefreshStrategy.StopRefreshAge` setting.

Note

The Social Connected module does not provide message statistics for social marketer messages.

Chapter 5

Analytics Features

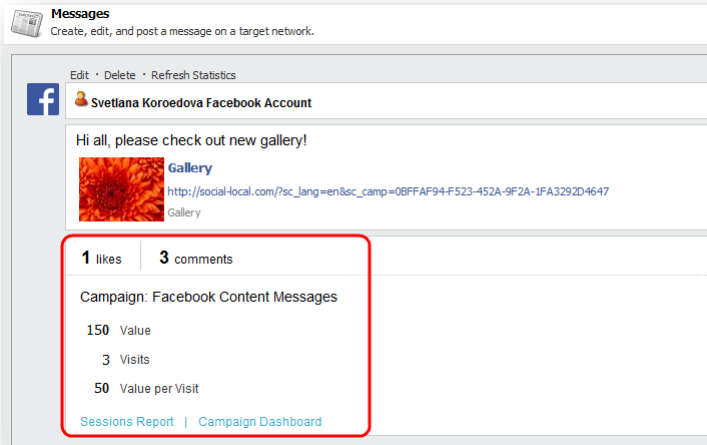
You can use the Social Connected module and the Sitecore Digital Marketing Suite to personalize your website. The Social Connected module provides the Like and Tweet buttons as standard Sitecore controls that track visitors' activity.

This chapter contains the following sections:

- Tracking Campaign Statistics on a Content Message
- Putting Share Buttons on the Web Page
- Tracking Share Buttons Usage
- Personalization Rules for the Social Connected Module
- Tracking Visits from Social Networks

5.1 Tracking Campaign Statistics on a Content Message

If you use the Social Connected module with DMS, you can track the campaign statistics on a content message.



These statistics are available for the messages that have campaigns assigned.

To see the campaign statistics, click **Show campaign reports**.

By default, a content message uses the generic *Twitter Content Messages* or *Facebook Content Messages* campaigns. The reports for these campaigns give you an overview of the effectiveness of all your Twitter or Facebook messages.

To only see the numbers for a particular message, when you set up the message, click **Create campaign** to create a new campaign for this message. For more information about creating a message, see the section, *Creating a Message*.

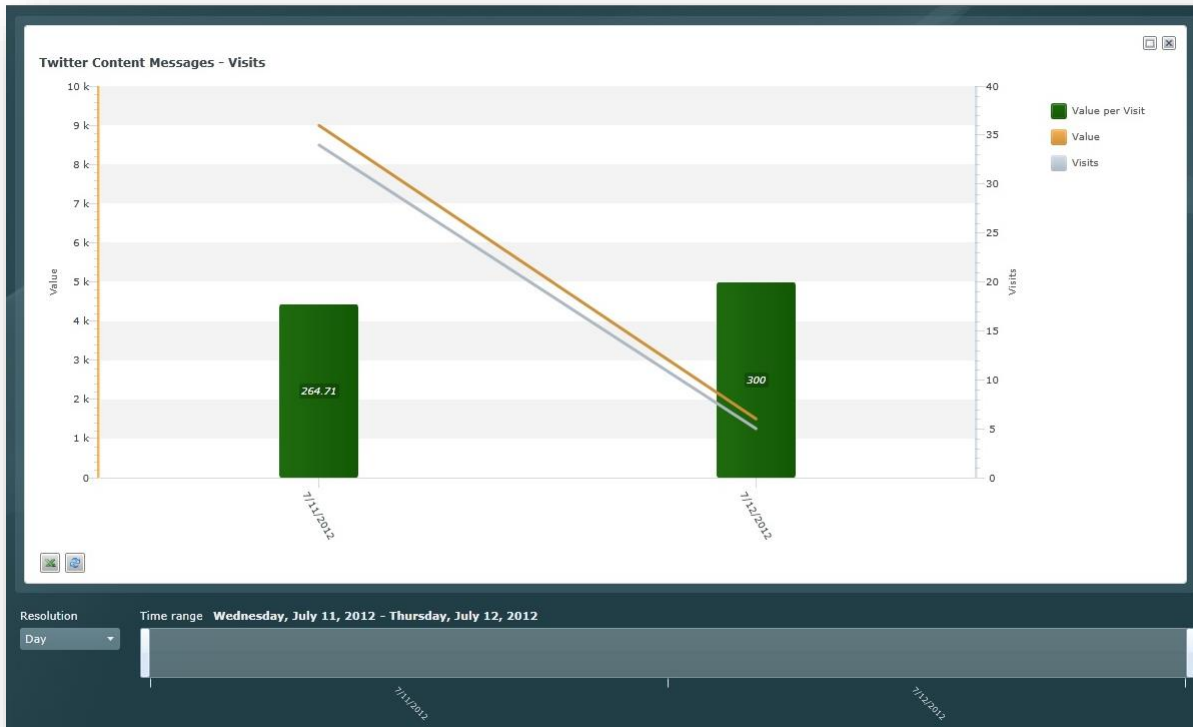
To get detailed information about a session report or a campaign dashboard, click the respective link.

Note

The Social Connected module gathers campaign statistics from DMS aggregate tables. As a result, the campaign statistics are updated as often as the DMS aggregation task runs.

Campaign Dashboard report

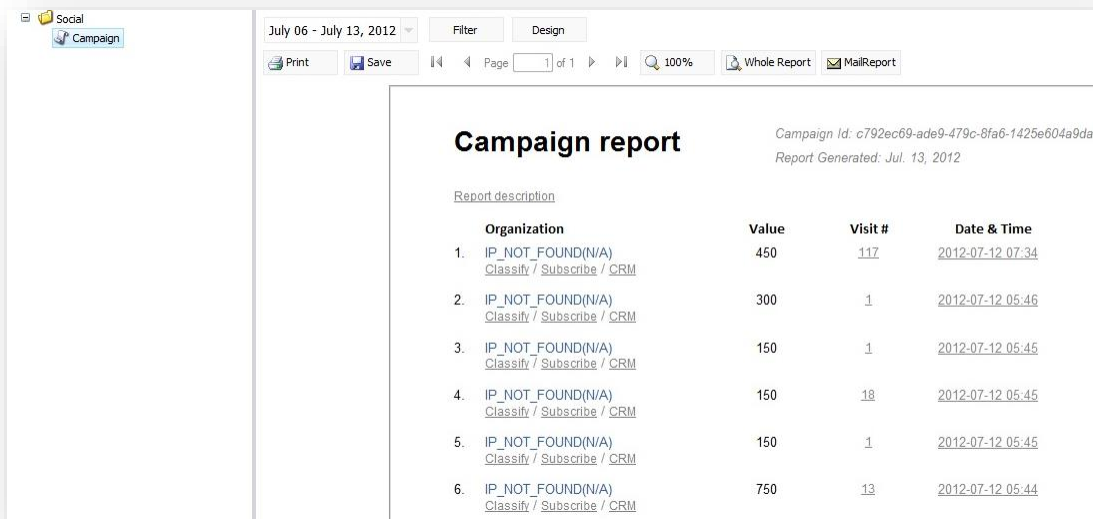
Campaign Dashboard report presents a campaign statistics overview, data over time, using graphs.



Session report

Session report presents detailed campaign information, along with review of individual sessions and visits to understand more about the experience of the visitors after they click the twitter/facebook link. This

report shows what visitors are doing on the website, what are potential problems in their experience.



The screenshot shows a 'Campaign report' interface. At the top, it displays the date range 'July 06 - July 13, 2012' and a 'Filter' button. Below this are navigation controls including 'Print', 'Save', 'Page 1 of 1', '100%' zoom, 'Whole Report', and 'MailReport'. The main content area is titled 'Campaign report' and includes the following information:

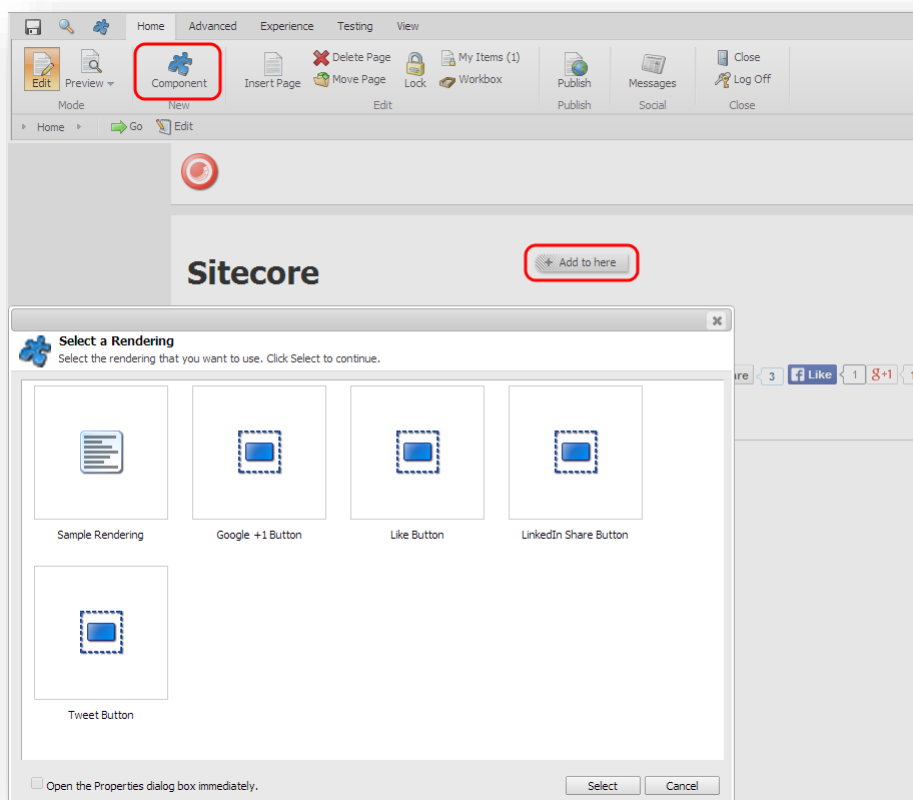
- Campaign Id: c792ec69-ade9-479c-8fa6-1425e604a9da
- Report Generated: Jul. 13, 2012
- Report description

	Organization	Value	Visit #	Date & Time
1.	IP_NOT_FOUND(N/A) Classify / Subscribe / CRM	450	117	2012-07-12 07:34
2.	IP_NOT_FOUND(N/A) Classify / Subscribe / CRM	300	1	2012-07-12 05:46
3.	IP_NOT_FOUND(N/A) Classify / Subscribe / CRM	150	1	2012-07-12 05:45
4.	IP_NOT_FOUND(N/A) Classify / Subscribe / CRM	150	18	2012-07-12 05:45
5.	IP_NOT_FOUND(N/A) Classify / Subscribe / CRM	150	1	2012-07-12 05:45
6.	IP_NOT_FOUND(N/A) Classify / Subscribe / CRM	750	13	2012-07-12 05:44

For more information about DMS reports, see the documentation on SDN.

5.2 Putting Share Buttons on the Web Page

You can put the Like (for Facebook), Tweet (for Twitter), +1 (for Google+) and Share (for LinkedIn) buttons on the website.



Sitecore provides them as standard controls. You can find the controls in the `sitecore/layout/sublayouts/social/sharing` folder. For the MVC layout, use the Social Connected renderings in the `sitecore/layout/Renderings/Social MVC/Sharing` folder.

To make sure that users can put these controls on a webpage, you must enable the controls on the appropriate placeholder:

1. In the **Content Editor**, navigate to `sitecore/layout/placeholder settings` folder and select the placeholder item where the controls will be located
2. In the **Allowed Controls** field, add the `Like Button` and the `Tweet Button` controls to the list of the selected controls.

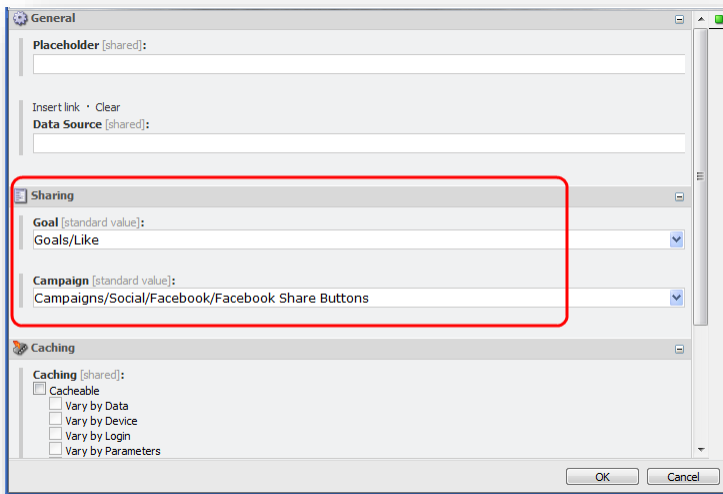
Note

For the Tweet share button to work correctly, in the `website\layouts\system\social\sharing\TweetButton.aspx` file, make sure that the `data-url` and `data-counturl` attributes have the same value.

5.3 Tracking Share Buttons Usage

After you placed the share buttons on the website, you can assign campaigns and goals to these buttons. Every time a website visitor clicks the button, a specific goal or campaign is triggered and registered in Sitecore Analytics.

The Social Connected module provides *Like*, *Tweet*, *Google +1* and *LinkedIn Share* goals for the share buttons. You can find these goals in the `sitecore/system/marketing center/goals` folder. Make sure you configured the goals before using them.



The module also provides several campaigns specific to social network activities. You can find social campaigns in the `sitecore/system/marketing center/campaigns/social` folder.

For more information about goals and campaigns, see the manual *Marketing Operation Cookbook* on the SDN.

Note

Changing the campaign assigned to the Tweet share button resets the tweet counter to null.

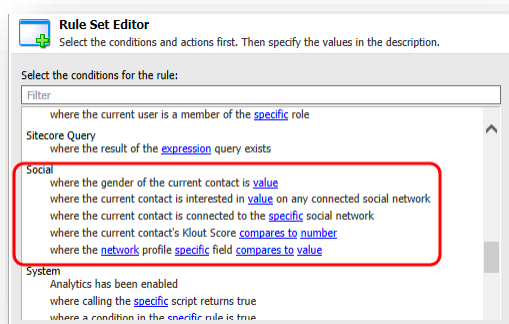
5.4 Personalization Rules for the Social Connected Module

The Social Connected module adds five personalization conditions and an action to the Sitecore Rule Set Editor.

You can use these personalization conditions and the action with the Social Connected module to personalize the website.

The Social Connected module provides the following conditions:

- *where the gender of the current contact is [value](#)*
- *where the current contact is interested in [value](#) on any connected social network*
- *where the current contact is connected to the [specific](#) social network*
- *where the current contact's Klout Score [compares to number](#)*
- *where the [network](#) profile [specific](#) field [compares to value](#)*



The Social Connected module provides the following action:

- *send associated messages.*

For more information about personalization rules, see the manual *Marketing Operation Cookbook* on the SDN.

Where the gender of the current contact is value

This condition defines visitor's gender.

Where the current contact is interested in value on any connected social network

This condition checks for the existence of the value in multiple profile fields in each connected network.

Editors can use this rule to define visitors interested, for example, in sports, instead of creating multiple rules to check individual profile attributes, such as interests, hobbies, books, and so on.

The set of attributes that this condition checks is configured in the `social.network_name.config` file, in the `<interestedInRule>` section.

Note

The *where the current user is interested in value on any connected social network* condition only supports one keyword for the value. Consider creating several conditions to use several values.

Where the current contact is connected to the specific social network

This condition checks whether the visitor is connected to the specific social network.

Where the current contact's Klout Score compares to number

This condition compares the contact's Klout Score with the expression set.

Where the network profile specific field compares to value

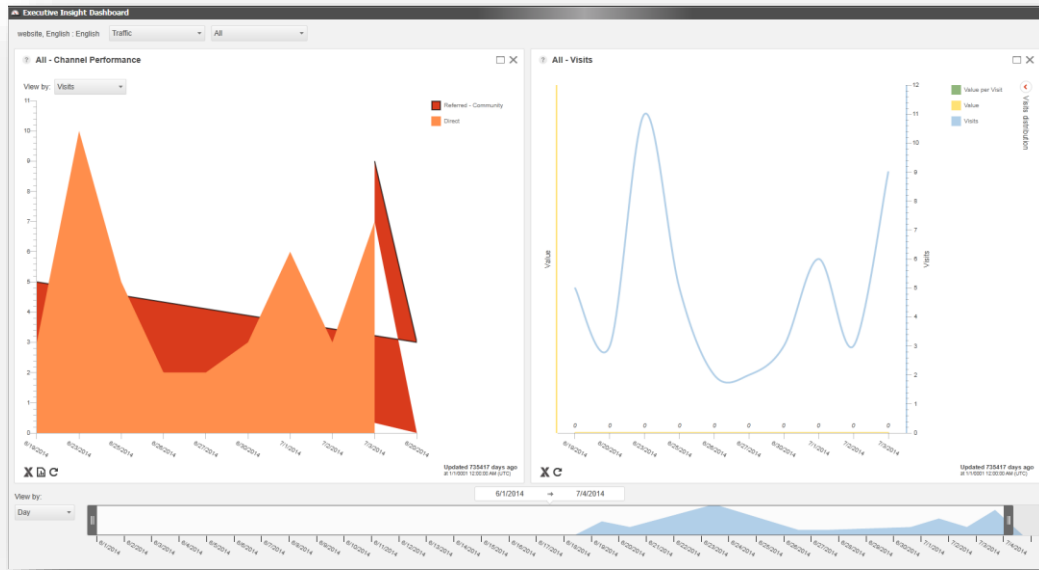
This condition checks whether a specific profile field contains a specific value.

Use this condition when you need more specific requirements that are not covered by other rules.

5.5 Tracking Visits from Social Networks

Sitecore Analytics registers incoming visitors' traffic. The Executive Insight Dashboard provides this information grouped by traffic types. All the visitors who come to your website from Facebook, Twitter, LinkedIn, and Google+ using reference links in the messages are stored in the **Referred - Community** traffic type.

You can assign visitors from any other social network to this traffic type in the Executive Insight Dashboard.



For more information about how to work with the Executive Insight Dashboard, see the *Executive Dashboard Cookbook* document.

Chapter 6

Developer's Notes

This chapter contains the advanced configuration information.

This chapter contains the following sections:

- Configuring the Module on Scaled Environment
- Working with Detached Messages
- The Social Connected Module Pipelines
- Configuring the Messages Search
- The Social Connected Events

6.1 Configuring the Module on Scaled Environment

This section describes how to configure the Social Connected module to work on a scaled environment.

6.1.1 Multi-server Environment

You can scale the Social Connected module by configuring a multi-server environment.

To configure a multi-server environment:

1. Install the Social Connected package on Content Management (CM) and Content Delivery (CD) servers. For more information about installing the module, see *Installation Guide* on the SDN.
2. Configure all the servers.

Configuring the servers

When you have installed the module on all the servers, you must configure them using the following instructions:

On all your CD servers:

1. In the `\Website\App_Config\Include\Social` folder, find all the configuration files with the `Index.Master` suffix in the name.
2. Rename these files to `[file_name].config.disabled`

On CD and CM servers:

1. In the `\Website\App_Config\Include\Social` folder, rename the `Sitecore.Social.ScalabilitySettings.config.disabled` file to `Sitecore.Social.ScalabilitySettings.config`.
2. In the file, set the value of the `Social.Scalability.ContentManagement.TasksExecutionInstance` setting to the name of the CM server that performs all the scheduled activities.
For information about renaming the instance, see the manual *Scaling Guide*, the section, *How to: Configure the name of the publishing instance*.

On all CM servers:

1. In the `\Website\App_Config\Include\Social` folder, find the `Sitecore.Social.config` file.
2. Set the value of the `Social.LinkDomain` setting to URL of the CD server.

On all the servers except for the CM server that performs scheduled activities (optional):

1. In the `\Website\App_Config\Include\Social` folder, find the `Sitecore.Social.config` file.
2. In the `scheduling` section, for all four agents, set the interval parameter to `00:00:00`.
3. If you have enabled `Sitecore.Social.Klout.config`, in the `scheduling` section, in the `Sitecore.Social.Klout.Client.Tasks.SynchronizeKloutScores` agent, set the interval parameter to `00:00:00`.

6.2 Working with Detached Messages

A detached message is a message that is not bound to a content item. You can create, edit, and post detached messages using API. Once the message is posted to the social network, you can track its campaign statistics using API.

For more information about the Social Connected API, see the document, *Social Connected API*.

6.3 The Social Connected Module Pipelines

The Social Connected module uses six pipelines.

6.3.1 The social.CreateMessage Pipeline

The `social.CreateMessage` pipeline is used when the module creates a message for a social network.

GetRoot Processor

The `GetRoot` processor gets a path in the content tree for the message item.

CreateItem Processor

The `CreateItem` processor creates two items under the messages root item: message and source items.

SaveData Processor

The `SaveData` processor saves message data from its fields.

6.3.2 The social.ReadMessage Pipeline

The `social.ReadMessage` pipeline reads all the messages for the current item from the content tree and returns the messages collection.

ReadMessages Processor

The `ReadMessages` processor reads all the messages for the specified message IDs.

6.3.3 The social.PostMessage Pipeline

The `social.PostMessage` pipeline posts the message to the social network.

ResolveMessagePostingProvider Processor

The `ResolveMessagePostingProvider` processor selects a posting provider to post the message.

PostMessage Processor

The `PostMessage` processor posts the message to the social network.

6.3.4 The social.BuildMessage Pipeline

The `social.buildMessage` pipeline builds the message before it is posted to the social network. It returns the message object that is ready to be posted.

ResolveRenderer Processor

The `ResolveRenderer` processor returns the message builder for the posting configuration (goal or content).

BuildMessage Processor

The `BuildMessage` processor builds the message.

ReplaceTokens Processor

The `ReplaceTokens` processor replaces tokens with their values.

6.3.5 The `social.MatchUser` Pipeline

The `social.MatchUser` pipeline uses the data provided by the user to search for an existing user or creates a new user if no existing user is found.

PrepareUserData Processor

The `PrepareUserData` processor creates data for the user: username and email. This is based on the user basic data.

FindByEmail Processor

The `FindByEmail` processor uses the email information provided by the user to search for an existing user.

FindByDomain Processor

The `FindByDomain` processor searches for an existing user by domain.

FindByNetworkCredentials Processor

The `FindByNetworkCredentials` processor uses the network credentials (ID) to search for an existing user.

CreateUser Processor

The `CreateUser` processor creates a new user.

6.3.6 The `social.InitializeApi` Pipeline

The `social.initializeApi` pipeline initializes social API.

IoCInitialization

The `IoCInitialization` processor initializes the inversion of control (IoC) container.

6.4 Configuring the Messages Search

The Social Connected module stores messages in the `Sitecore/Social/Messages` folder. This folder is an item bucket that has a configured index. The Social Connected module uses the index to search for the messages. This index is based on the Lucene or Solr provider, and the `Sitecore.ContentSearch` API.

For more information about how to configure the search functionality, see the *Sitecore Search and Indexing Guide*.

In the following table, you can see the files, in which you can change the index configuration for Lucene. The configuration files for Solr are identical but contain Solr in the name of the files instead of Lucene.

Configuration file	Section	Description
<code>[website_root]\app_config\Include\social\Sitecore.Social.Lucene.IndexConfiguration.config</code>	<code>social/indexConfigurations</code>	In this section, you can edit the Lucene index configuration.
<code>[website_root]\app_config\Include\social\Sitecore.Social.Lucene.Index.Master.config</code>	<code>/sitecore/contentSearch/configuration/indexes/index</code>	This section contains the index configuration for the Content Management server. The index uses the <code>syncMaster</code> strategy and is updated when any item in the <code>Message</code> folder is updated. The Sitecore job updates the index according to a schedule, which leads to some delays in updating the index.
<code>\Website\App_Config\Include\social\Sitecore.Social.Lucene.Index.Web.config</code>	<code>/sitecore/contentSearch/configuration/indexes/index</code>	This section contains the index configuration for the Content Delivery server. The index uses the <code>OnPublishEndAsynch</code> strategy and is updated immediately when you publish items to a Content Delivery server.

`Sitecore.social.config` contains the following settings that are common for both Lucene and Solr indexes:

- `Social.MessagesByContainerCache.Enabled`
If this setting is set to *True*, the Social Connected module caches the index values found inside a goal item, a content item, or an empty container. If this setting is set to *False*, to see the changes in the messages, in the **Messages** dialog box, click the **Refresh** button. You can change the cache size in the `Social.MessagesByContainerCache.Size` setting.

Note

The cache helps you to get the latest indexed data inside a container. Other API methods, for example, `GetMessagesByWorkflowState` or `GetMessagesByAccount`, depend on the corresponding job execution.

If you install the Social Connected module package on the Content Delivery server, you must edit the `\Website\App_Config\Include\social\Sitecore.Social.Lucene.Index.Web.config` or `\Website\App_Config\Include\social\Sitecore.Social.Solr.Index.Web.config` file

specifying the name of the Content Delivery server database in the `database` section. If you use one common server for the Content Management and Content Delivery servers with the Master and Web databases, you can use the default settings.

If you create a custom index for the Content Management or Content Delivery server, you must specify the index ID in the corresponding settings:

- `Social.Messages.SearchIndex.Master` – the name of the Content Management server index.
- `Social.Messages.SearchIndex.Web` – the name of the Content Delivery server index.

6.4.1 Solr Configuration

Sitecore uses Solr, which contains a single core named `itembuckets`. To make the Social Connected module work correctly with Solr, you must create a new core in Solr named `social`. If you give the new core a different name, you must specify this name in the `Sitecore.Social.Solr.Index.Master.config` and `Sitecore.Social.Solr.Index.Web.config` files, in the `param` section.

For more information about how to configure Solr, see the Solr documentation.

6.5 The Social Connected Events

The Social Connected module events let you process activities related to Social Connector and to messages. The `Sitecore.Social.config` file contains the events; the `Sitecore.Social.ScalabilitySettings.config` file contains the remote events.

Event	Description
<code>social:connector:user:created</code>	A new Sitecore user has been created by the module.
<code>social:connector:user:created:remote</code>	A new Sitecore user has been created on a remote Sitecore instance by the module.
<code>social:connector:user:loggedin</code>	A user has logged into the website using social network credentials.
<code>social:connector:user:loggedin:remote</code>	A user has logged into the website on a remote Sitecore instance using social network credentials.
<code>social:connector:user:socialprofileattached</code>	A user has attached a new social profile to their current profile.
<code>social:connector:user:socialprofileattached:remote</code>	A user has attached a new social profile to their current profile on a remote Sitecore instance.
<code>social:message:created</code>	A social message has been created.
<code>social:message:created:remote</code>	A social message has been created on a remote instance.
<code>social:message:deleted</code>	A social message has been deleted.
<code>social:message:deleted:remote</code>	A social message has been deleted on a remote instance.
<code>social:message:updated</code>	A social message has been updated.
<code>social:message:updated:remote</code>	A social message has been updated on a remote server.

To disable a remote event, comment the handler that calls this event in the `Sitecore.Social.ScalabilitySettings.config` file.

For example, to disable the `social:message:created:remote` event, in the `Sitecore.Social.ScalabilitySettings.config` file, comment this code:

```
<event name="social:message:created">
  <!-- Raises the "social:message:created:remote" event. -->
  <!--<handler
type="Sitecore.Social.Client.MessagePosting.Handlers.SocialMessageCreatedHandler,
Sitecore.Social.Client" method="OnSocialMessageCreated"/> -->
</event>
```

Chapter 7

Tips and Tricks

This chapter contains some tips and tricks for developers and administrators. This chapter contains the following sections:

- The Security Roles Used in the Social Connected Module
- How to Use the Error Handling Mechanism
- Configuring Advanced Logging

7.1 The Security Roles Used in the Social Connected Module

In the following table, lists the security roles for content authors and reviewers:

Role	Description	Requires
sitecore\Author	Permissions to the Content Editor, Workbox and Messages dialog box	Author (required) Reviewer (required)
sitecore\Sitecore Client Social Authoring	This role is a member of the <i>Sitecore\Author</i> role. Permissions to the Messages button on the Social chunk of the Publish ribbon.	Author (required) Reviewer (required)
sitecore\Analytics Maintaining	Permissions to create goal/page event messages and campaigns for messages.	Author (optional)
sitecore\Analytics Reporting	Permissions to view DMS reports for a message	Author (optional)
sitecore\Social Message Author	Permissions to create, edit, and remove all messages	Author (required)
sitecore\Facebook Message Reviewer	Permissions to edit Facebook messages	Reviewer (required)
sitecore\Twitter Message Reviewer	Permissions to edit Twitter messages	Reviewer (required)
sitecore\Social Marketer Message Reviewer	Permissions to edit social marketer messages	Reviewer (required)
sitecore\Social Message Workflow Editor	Permissions to use the <i>Draft</i> state and its commands of the <i>Social Message</i> workflow and <i>Social Marketer Message</i> workflow	Author (required)
sitecore\Social Message Workflow Reviewer	Permissions to use the <i>Awaiting Approval</i> state and its commands of the <i>Social Message</i> workflow. Permissions to use the <i>Awaiting Post Review</i> , <i>Post Accepted</i> and <i>Post Removed</i> states and their commands of the <i>Social Marketer Message</i> workflow	Reviewer (required)

7.2 How to Use the Error Handling Mechanism

The Social Connected module contains an error handling mechanism that you can use to display error messages both on the website and in the Sitecore Desktop. This mechanism lets you personalize the look and style of the errors that are displayed in your solution.

The error handling mechanism deals with the module features that directly cooperate with social networks. An error is typically comprised of a simple message and a detailed message. You can find the detailed message in the module log files under the folder `[installation_root]\data\logs`. The simple message is displayed in the UI.

You can use the language packages provided with the module to localize the simple error messages.

7.2.1 Error Handling on the Website

Any errors that are caused by visitors on the website when they use the Social Connector log in buttons are also registered. The module registers detailed error messages in the log files.

To display the error messages to website visitors in the style used by your website, a developer should follow this procedure:

1. Create a control that looks for authentication errors in the query string and gets the GUID of the error.
2. Get the simple and detailed error message using exception manager and method

```
var exceptionClientManager = new ExceptionClientManager();
    var exceptionInfo =
exceptionClientManager.GetExceptionInfo(HttpContext.Current.Request);
    if (!exceptionInfo.IsEmpty)
    {
        this.ShowExceptionInfo(exceptionInfo);
    }
```

3. Display this information to the website visitors in the same style as the website design.

7.3 Configuring Advanced Logging

To enable registering Social Connected tracing information in the log files:

1. Edit the `Sitecore.Social.config` file.
2. Set the following setting to true:

```
<setting name="Social.Logging.TraceToLog" value="true">
```

Using the Social Connected API, you can register information about objects, static classes, and error messages.

An object type or a static class must have an appropriate formatter that logs either objects of this type or the type itself. To customize logging, specify the assemblies in which the formatters are stored:

1. Edit the `Sitecore.Social.config` file.
2. In the `entityFormatterAssemblies` node, in the `assembly name` parameter, specify the DLL in which the module searches formatters:

```
<logging>  
  <!-- Assemblies, used to search entity formatters for logging. -->  
  <entityFormatterAssemblies>  
    <assembly name="Sitecore.Social.Infrastructure.Logging.dll" />  
  </entityFormatterAssemblies>  
</logging>
```

Formatters are classes that implement the `Sitecore.Social.Infrastructure.Logging.IEntityFormatter` interface. Formatters are decorated with the `Sitecore.Social.Infrastructure.Logging.FormatterForAttribute` attribute that specifies the type to log.

For the types marked with the `System.SerializableAttribute` attribute, if there is no formatter defined for a type, the system uses the default one.

For more information about how to use the API for logging, see the `Sitecore.Social.Infrastructure.Logging` namespace in the *Social Connected API* document.